MULTIFACILITY LOCATION PROBLEMS ON SOME SPECIAL NETWORKS

BY

YANG XU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1993

ACKNOWLEDGEMENTS

I have long been looking forward to this opportunity to express my deep gratitude to Professor R. L. Francis, the chairman of my supervisory committee. Professor Francis not only directed me to the research topics that culminate in this dissertation, and provided numerous valuable suggestions, but also gave me precious advice in regard to the principles of conducting research, giving presentations, scholarly writing, and academic teaching. The amount of time, the thoroughness, and the quick and valuable feedback in reading and correcting many of my dissertation drafts demonstrate his professional dedication and his great patience toward his students. Without his efforts this work would not be possible.

I thank the other members of my supervisory committee, Professors H.P. Benson, D.J. Elzinga, D.W. Hearn, and C.Y. Lee, for their time in reading my dissertation. I give special thanks to Professors Lee for his encouragement in all these years.

I thank Professor Tim Lowe of the University of Iowa, who had played an important role in initiating the research in this dissertation and made several contributions that led to the conciseness and clarity of some of the results in this dissertation.

I am grateful to the Department of Industrial and Systems Engineering of the University of Florida for financially supporting me throughout my graduate studies. Part of my research was suported by NSF grants ECS-8612907 and ECS-8612911.

Finally, I would like to express my great appreciation to my wife, Weili, for her support and understanding.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School of the University of Florida in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

MULTIFACILITY LOCATION PROBLEMS ON SOME SPECIAL NETWORKS

By

Yang Xu

August 1993

Chairman: Dr. Richard. L. Francis
Major Department: Industrial and Systems Engineering

In this dissertation we develop theories and algorithms for some multifacility location problems which, as a class, are to determine the locations, on a transportation network, of a set of functionally distinct facilities, to minimize an objective function of demand-point to facility distances and inter-facility distances.

Unlike the cases when the network is a tree or the distances are rectilinear, all the problems are provably difficult when the network contains cycles. Thus, we focus on two special cyclic networks -- the multiblock network and the grid network — which are, respectively, generalizations of a tree network and a rectilinear grid.

For the multimedian problem on a multiblock network, we develop a method that, in polynomial time, localizes every new facility to either a vertex or a block. With the localization, the problem can be decomposed.

For the multimedian problem on a grid network, we develop a branch and bound algorithm with the rectilinear multimedian problem as a lower bounding problem. We also develop a polynomial-time algorithm for an approximation. Computational experiments are conducted.

Finally, we propose a branch and bound approach for a class of multifacility location problems on general cyclic networks, and a special approach for the same problems on grid networks. With polytope-type solution subsets and with piecewise linear and convex underestimates for the shortest distances, the lower bounding problems are convex if the problem objective function is a convex function of distances; and the lower bounding problems are linear in many special cases. The distance underestimates approach the originals as the level of decomposition increases. When the network is a grid network, we derive specialized lower bounding problems with substantially improved quality.

CHAPTER 1
INTRODUCTION

Location decisions involve various spatial resource allocation problems in which a set of existing facilities is spatially distributed over a region. One needs to determine the locations of new facilities in the area and/or to allocate existing facilities to new facilities to optimize some objective. The objective is usually a real-valued function of distances between pairs of existing/new facilities (Type I distances) and/or between pairs of new facilities (Type II distances). A network location problem occurs when the point to point traffic must follow a pre-specified network (e.g. a road network). Every existing facility is a vertex in the network and the new facility locations must be on the network. Usually, the distances involved in the objective are the network shortest path distances.

In this dissertation, we consider network multifacility location problems with mutual communication (multifacility problems for short), which are to determine the locations of a set of distinct new facilities with the objective functions depending on both types of distances. The applications of multifacility problems involve those which consider the locations of different types of new facilities (especially, new facilities with hierarchies). The examples include determining the locations of some function-specified subsidiaries of a company (warehouses and plants, work stations on a workshop floor); determining the locations of some service centers (hospitals, clinics, ambulance stations); the locations of computing resources and information storage units on a distributed computer network. Some multifacility location problems are also closely related to facility layout problems.

The focus of this dissertation is on two particular problems — the multimedian problem (the minisum multifacility problem or the p-median problem with mutual communication) and the multicenter problem (the minmax multifacility problem or the p-center problem with mutual communication). Both problems are NP-hard on general networks and have known polynomial-time algorithms only when the underlying networks are trees. Our objective is to design improved algorithms for both problems when defined on networks more general than tree networks and gain insight for the problems under the most general assumptions.

1.1. Definitions and Notation

In this section we will define the multimedian problem and the multicenter problem, and introduce notation.

An undirected network $G = (V, E)$ has a vertex set $V = \{v_1, \ldots, v_m\}$ and an edge set $E = \{(v_i, v_j) : \text{for some } v_i, v_j \text{ in } V\}$. Each edge has a positive length. We always assume that G is connected. If a point lies in the interior of an edge it is an interior point and is represented by its distances to the vertices which are the end points of the edge containing the interior point. For any two points x, y of G, we denote $d(x, y)$ as the shortest distance between them. It is well-known that $d(.\,,.)$ is a metric having the following properties: (i) (nonnegativity) $d(x, y) \geq 0$, with $d(x, y) = 0$ if and only if $x = y$; (ii) (symmetry) $d(x, y) = d(y, x)$; (iii) (triangle inequality) $d(x, y) \leq d(x, z) + d(z, y)$ for any $z \in G$. A network is a tree network if it is connected and has no cycles. For a more rigorous definition of a network, see Dearing et al. (1976).

Unless otherwise stated, we always assume that a multifacility problem has m existing facilities and n new facilities. Let $J \equiv \{1, ..., n\}$ be the index set of new facilities. In a location network, each vertex v_i is an existing facility with nonnegative weight w_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$. There is also a nonnegative weight v_{jk} for new facilities j and k where $v_{jk} = 0$ if $j = k$, and $v_{jk} = v_{kj}$ for all j and k. A weight describes the interaction intensity between the associated pair of facilities.

A multimedian problem is defined as the following:

$$\text{MMP: } \underset{X \in G^n}{\text{Minimize}}\ f(X) \equiv \sum_{j=1}^{n} f_j(x_j) + f_{NN}(X),$$

where G^n is the n-fold Cartesian product of G, $f_j(x_j) = \sum \{w_{ij}\, d(v_i, x_j) : i = 1, \ldots, m\}$ $j \in J$, and $f_{NN}(X) = \sum\sum \{v_{jk}\, d(x_j, x_k) : 1 \le j < k \le n\}$.

The multicenter problem differs from the multimedian problem in the objective:

MCP: $\underset{X \in G^n}{\text{Minimize}}\ f(X) \equiv \max\{\ \max\{f_j(x_j): j \in J\},\ f_{NN}(X)\ \}$, where

$f_j(x_j) = \max\{w_{ij}d(v_i, x_j) \mid i = 1, \ldots, m\}$, $j \in J$ and now $f_{NN}(X) = \max\{v_{jk}d(x_j, x_k) \mid \text{for all } j < k\}$.

For completeness, we include, in both problems, the single facility case (n = 1) where there is no $f_{NN}(X)$ term. When n = 1, MMP and MCP are, respectively, the well-known 1-median problem and 1-center problem (Hakimi 1964). In both MMP and MCP, it is the interaction terms ($v_{jk}d(x_j, x_k)$) that bind the problem together. Let $G_I = (J, E_I)$ be the interaction graph such that an edge (j, k) is in E_I if and only if $v_{jk} > 0$. We always assume that G_I is connected since otherwise the corresponding problem can be decomposed into several independent problems, each of which corresponds to a component of G_I.

1.2 Literature Review

In this section, we give a general literature review of the topics related to this dissertation. In the first subsection, we review past work on multifacility location problems. In the second subsection, we review the literature of location problems on special cyclic networks.

1.2.1 A General Review

After nearly 30 years of active development, location theory has grown into many branches. A recent survey (Brandeau and Chiu, 1989) lists 58 major location problems. Here, we concentrate only on the literature on multifacility location problems. For a general survey one can refer to the following: Francis et al. (1983), Tansel et al. (1983a), Krarup and Pruzan (1979, 1983), Aikens (1985), Brandeau and Chiu (1989). Furthermore, there are text books by Francis, McGinnis, and White (1992), by Handler and Mirchandani (1979), and by Love et al. (1988). A recent book edited by Mirchandani and Francis (1990) discusses discrete location problems. For some location-related research in disciplines other than operations research, one can refer to the book edited by Ghosh and Rushton (1987).

The planar multifacility location problem developed much earlier than its network counterpart and has attracted many researchers. Most of the research on the planar multifacility location problem concentrates on problems with Euclidean distances (Rado, 1988) or rectilinear distances (Francis, McGinnis, and White, 1992). Also, there is an extensive discussion on problems with l_p distances (Hansen and Thisse, 1983; Idrissi et al. 1987). In this proposal, our main interest is the rectilinear multifacility location problem, due to its relation to grid networks.

If the distances are rectilinear distances in a multimedian problem, the problem is called the rectilinear multimedian problem. A rectilinear multimedian problem can be decomposed into two independent subproblems, each of which is a multimedian problem defined on a path (a special tree network). All the known algorithms use this fact. Cabot et al. (1970) found an equivalent linear program for each subproblem and solved the dual of the linear program as a minimal cost flow problem. Picard and Ratliff (1978) gave a direct search algorithm of $O(mn^3)$ for each of the subproblems. The algorithm solves a minimum-cut problem on each edge of the path where the cut gives an optimal partition of new facilities over the two subtrees obtained by removing the edge. Based on a different point of view, Kolen (1981) gave a direct search algorithm which is equivalent to Picard and Ratliff's algorithm. Both of these direct search algorithms can be easily extended to the multimedian problem on tree networks and Kolen (1986) gave an explicit description of such an algorithm. For other early work on the rectilinear multimedian problem, see Pritsker and Ghare (1970), Rao (1973), Juel and Love (1976) and Sherali and Shetty (1978).

No known method exists to decompose the rectilinear multicenter problem directly. Instead, there is a one-one mapping between rectilinear distance space and Tchbyshev distance space; the multicenter problem defined with Tchbyshev distance can be decomposed into two independent subproblems. With this decomposition, Dearing and Francis (1974) formulated the subproblems as linear programming problems and solved the duals as special network flow problems. Similar to the rectilinear multimedian problem, each subproblem of the rectilinear multicenter problem can be viewed as a multicenter problem on a path. Hence, the solution techniques on a tree network can be applied here.

Multimedian and multicenter network location problems are defined by Dearing et al. (1976) under the presence of distance constraints. Since then, most of the research has concentrated on the problems defined on tree networks due to the fact that each problem is convex if and only if the network is a tree (Dearing et al. 1976), and each problem is NP-hard on general cyclic networks (Kolen 1982).

Research on the 1-median problem and the 1-center problem is vast. For the 1-median problem, Hakimi (1964) showed that there is an optimal solution at a vertex (Vertex Optimality Property). For the problem on a tree network, Goldman (1971) and Lo-Keng Hua et al. (1962) independently gave a tree trimming algorithm of O(n), which is based on a "Majority Localization Condition" (Goldman and Witzgall, 1970).

For the 1-center problem on a general network, Hakimi (1964) showed that the candidates for optimal solutions can be a finite set, namely the set of vertices and <u>bottleneck intersection points</u>. Based on this, Kariv and Hakimi (1979b) gave an $O(|E|n\log(n))$ algorithm. For the unweighted 1-center problem on a tree network, Goldman (1972) gave an $O(n^2)$ algorithm using the fact that at any vertex one can always tell which <u>induced subtree</u> contains an optimal solution. Dearing and Francis (1974) showed that $\max\{w_i w_j d(v_i, v_j)/(w_i+w_j): 1 \leq i < j \leq m\}$ is a lower bound to the 1-center problem when the network is cyclic and the bound is tight when the network is a tree. Kariv and Hakimi (1979b) gave an $O(n\log(n))$ algorithm for the tree 1-center problem.

The multimedian problem also has a vertex optimality property (Tansel et al., 1983b). Therefore, by removing an edge from a tree network, there must be an optimal partition of the new facilities over the two resulting subtrees. This partition is independent of the length of the edge removed (Kolen 1981). For the multimedian problem on a tree network (<u>tree multimedian</u>), Kolen (1981) gave a direct search algorithm similar to the one given by Picard and Ratliff (1978). The algorithm uses the convexity of the tree multimedian problem so that (i) a locally optimal solution is a globally optimal solution, and (ii) one can determine an optimal direction locally. It selects an arbitrary edge and solves a minimum cut problem on a flow network with n+2 nodes to determine a globally optimal partition of new facilities on the two subtrees obtained by removing the edge. The problem can then be decomposed into two subproblems. The algorithm continues to decompose until every subtree is a single vertex. The minimum cut set problem can be solved by Karzanov's maximum flow algorithm (1974) in $O(n^3)$, and the direct search algorithm solves a minimum-cut set problem on each of the edges in the tree, so that the direct search algorithm is $O(mn^3)$. The structure of the flow graph is also very important to the complexity of the multimedian problem. Chhajed and Lowe (1992) developed a polynomial-time algorithm when the flow graph is a series-parallel graph.

The results by Francis et al. (1978) and later the refined results by Tansel et al. (1980) and Erkut et al. (1989) represent another approach toward the multifacility problems on tree networks, especially under the presence of distance constraints. A network distance constraints problem is to find a solution for inequality system (DC): $D(X) \leq b$, which can be written explicitly as

$$d(v_i, x_j) \leq c_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n, \tag{1.1}$$

$$d(x_j, x_k) \leq b_{jk},\ 1 \leq j < k \leq n. \tag{1.2}$$

When G is a tree network T, Francis et al. (1978) gave the following results. Using the Neighborhood Intersection Procedure (NIP), one can construct a set of neighborhoods on T, $d(a_{(j)}, x_j) \leq c_j$, $j = 1, \ldots, n$, which is equivalent to (1.1). With the Sequential Location Procedure (SLP), of $O(m(m+n))$, one can either construct a feasible solution X on T or prove that $D(X) \leq b$ is inconsistent. An weight graph $N(b, c)$ consisting of m EF (existing facility) nodes EF_i, $i = 1, \ldots, m$ and n NF (new facility) nodes NF_j, $j = 1, \ldots, n$ is used. An edge (EF_i, NF_j) $((NF_j, NF_k))$ is in $N(b, c)$ with length c_{ij} (b_{jk}) if $c_{ij} > 0$ $(b_{jk} > 0)$. A path in $N(b, c)$ with two EF end nodes and only NF intermediate nodes is called a direct-path. Let $L(EF_i, EF_j{:}b, c)$ denote the shortest direct-path length in $N(b, c)$ between EF_i and EF_j. Then, (DC) is known to be consistent if and only if the following system is consistent:

$$L(EF_i, EF_j{:}b, c) \geq d(v_i, v_j),\ \text{for all i and j}. \tag{1.3}$$

The Separation Conditions (1.3) are equivalent to a linear inequality system $\mathbf{A}\,\mathbf{Z} \geq \mathbf{d}$ with $\mathbf{A}$ a direct-path vs. edge incidence matrix of $N(Z)$, $\mathbf{d}$ a distance vector with entries $d(v_i, v_j)$'s, and Z a vector with each entry some b_{ij} or c_{jk}.

For a tree network *T*, Erkut et al. (1989) discussed a general constrained multifacility problem with monotonic nondecreasing objective function. It can be formulated as:

P_1: Minimize $\{b_0 \mid X \in T, f_k(Z) \leq b_k, k = 0, 1, \ldots, p, D(X) \leq Z, Z \geq 0\}$.

From the Separation Conditions, $D(X) \leq Z$ is equivalent to $A\,Z \geq d$. Thus, to find an optimal vector Z^*, one can solve problem

P_2: Minimize $\{b_0 \mid f_k(Z) \leq b_k, k = 0, \ldots, p, \ A\,Z \geq d, \ Z \geq 0\}$

With an optimal Z^*, one can use algorithm SLP to find a corresponding optimal solution X^* on *T* by solving $D(X^*) \leq Z^*$.

If the functions $f_k(Z)$ are piecewise linear and non-decreasing, Erkut et al. (1989) showed that P_2 is equivalent to a linear programming problem, say LP_2, containing $A\,Z \geq d$ as a major part of constraints, and that the dual of LP_2 can be solved with an efficient column pricing procedure based on SLP. Our multimedian and multicenter problems are two special problems. Writing MMP in a vector form, we have

MMP: minimize $\{w^T Z \mid X \in T, D(X) \leq Z, Z \leq 0\}$.

The corresponding linear program is

MMP_L: minimize $\{w^T Z \mid A\,Z \geq d, Z \geq 0\}$.

For MCP, one can determine the optimal objective value z^* directly instead of by solving the corresponding linear programming problem (Francis et al. 1978). Let M be a large constant. Let

$$z_{ij}(E, N) = \begin{cases} z/w_{ij}, & \text{if } w_{ij} > 0, \\ M, & \text{o/w} \end{cases} \quad i = 1, \ldots, m, j = 1, \ldots, n, \qquad z_{jk}(N, N) = \begin{cases} z/v_{jk}, & \text{if } v_{jk} > 0 \\ M, & \text{o/w}, \end{cases} \quad 1 \leq j < k \leq n,$$

and let

$$b(z) = (z_{11}(E, N), \ldots, z_{mn}(E, N), z_{12}(N, N), \ldots, z_{n-1n}(N, N)),$$

A vector form of a MCP is MCP: minimize $\{z \mid X \in T, D(X) \leq b(z), z \geq 0\}$.

Erkut et al. (1992) considered finding z^* more efficiently through bisection search. Tansel et al. (1980) showed that the separation conditions provide more information. For a distance constraint system $D(X) \le b$, a direct path $P(EF_i, EF_j)$ in $N(b)$ is tight if $LP(EF_i, EF_j{:}b) = d(v_i, v_j)$. They showed that for any NF node in a tight path, its corresponding new facility is uniquely located using the solutions satisfying $D(X) \le b$. Thus, in MCP, z is the optimal objective function value if and only if $N(b(z))$ contains at least one tight direct path.

Finally in this subsection we discuss research on some cyclic network multifacility location problems. Due to the complexity of a cyclic network, both multimedian and multicenter problems become difficult. Dearing et al. (1976) pointed out that the tractability of most of the tree network location problems is partly due to the fact that on a tree network the type I and type II distances are all convex. It is well-known that in a cyclic network, the type I distance is piecewise concave on an edge. Hooker (1991) showed that in a cyclic network, the distance $d(x_j, x_k)$, for any $j \neq k$, is piecewise concave if x_j and x_k are in two different edges, and can be neither convex nor concave if x_j and x_k are on the same edge. Francis et al. (1978) showed that the separation conditions are only necessary conditions for a constraint system $D(X) \le b$ to be consistent if the network is cyclic. That is

$D(X) \le b$ is feasible $\Rightarrow L(EF_i, EF_j{:}b) \ge d(v_i, v_j),\ 1 \le i < j \le m \Leftrightarrow A\,Z \ge d$ is feasible.

Thus, for the multifacility problem P_1 above, the corresponding P_2 is a lower bounding problem (Erkut et al., 1989). Furthermore, even if one manages to obtain an optimal Z^* for P_1, it can be difficult to determine an optimal X^* since Kolen (1982) showed that the distance constraint problem is NP-hard on cyclic networks. On the other hand, let P_1' be a problem having the same set of data as P_1 but defined on some spanning tree of G, and let P_2' be the problem by replacing $D(X) \le Z$ in P_1' with $A\,Z \ge d_T$, where d_T is a vector of the distances between vertices on the spanning tree. Problem P_2' gives an optimal objective value of P_1' which is an upper bound on P_1. Erkut et al. (1988) report some experimental results on the quality of the lower bounds and upper bounds obtained in this way for the multimedian problem. It is worth noting that problem P_2 provides the only known nontrivial lower bound for multimedian and multicenter problems.

Realizing that a network edge can be decomposed into segments on which objective functions are convex for a class of location problems, Hooker (1986, 1989) gave algorithms for location problems with convex objective functions of type I distances. In the algorithms, subproblems are solved with respect to these segments.

1.2.2 Network Location Problems on Some Special Cyclic Networks

The fact that there are efficient algorithms for most of the tree network location problems and that most of the network location problems are NP-hard on general cyclic networks motivates the study of location problems defined on special cyclic networks. By introducing the concept of a gate and a gated subnetwork, Goldman and Witzgall (1970), and Goldman (1971) may be among the earliest to identify the reason why location problems on tree networks are so tractable. For a given subnetwork G' of G, let u be a point not in G'. Then a point g(u) in G' is the gate point of u if and only if for every point (or vertex) u' in G', g(u) is in every shortest path connecting points u and u' (i.e. d(u, u') = d(u, g(u)) + d(g(u), u')). A subnetwork G' is gated if for every point u (or vertex) not in G' there exists a **unique** gate point g(u) in G'. The uniqueness is important since it enables one to consider G' as an aggregated unit in some location problems. Goldman and Witzgall showed that for the 1-median problem, if the total weight of a gated subnetwork G' is at least half of the total weight of G, then G' contains an optimal location of the new facility. Note that any subtree of a tree network is gated; Goldman's tree trimming algorithm for the 1-median problem is based on this property. A subnetwork in a general cyclic network is very likely not gated (i.e. from some point outside the subnetwork, there are different gates to enter the subnetwork in order to reach different parts of the subnetwork via shortest routes.). Thus, one naturally intends to study location problems on special cyclic networks with easily identifiable gated subnetworks.

In this dissertation, we will study two special cyclic networks -- the multiblock network and the grid network. In the following, we define these two networks and discuss the related literature.

1.2.2.1. Location Problems on Multiblock Networks

First, we give the definition of a multiblock network. A cutpoint vertex of a graph is a one whose removal, together with the incident edges, increases the number of components. A graph is nonseparable if it is nontrivial, connected, and has no cutpoint vertices. A block is a maximal nonseparable subgraph (Harary 1969). A multiblock graph contains more than one block (has at least one cutpoint vertex). A network with a multiblock underlying graph is a multiblock network. Figure 1.1 shows a multiblock network with 3 cutpoint vertices and 4 blocks.

For the 1-median problem, by studying the weights of gated subnetworks, Chen et al. (1985) gave a polynomial time algorithm that either finds a vertex 1-median or localizes all the 1-medians to a single block. For the 1-center problem, Chen et al. (1988) gave an algorithm that gives similar localization results. Chang and Nemhauser (1982) considered a R-Domination problem on a multiblock graph. Gurevich et al. (1984) considered an r-covering problem on a multiblock network. For the r-covering problem, they gave an algorithm with complexity depending on the sizes of the blocks. Kim et al. (1989) considered a problem of locating a covering-type minimal-length-subgraph on a multiblock network. Based on that a 3-cactus network (a network with the underlying graph consisting of bi-connected cycles of three vertices) can be transformed into a tree network without changing the shortest distance between any pair of vertices, Kincaid and Lowe (1990) considered the 1-center problem on a 3-cactus network.

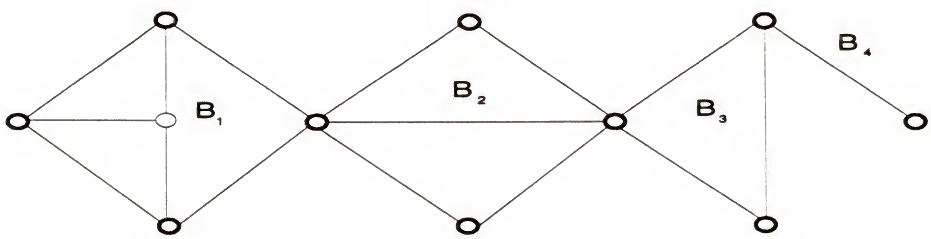


Figure 1.1 A Multi-Block Network

1.2.2.2. Location Problems on Grid Networks

The grid network is another kind of cyclic network with easily identifiable gated subnetworks. To our knowledge, there is no existing result for any location problem on a grid network that exploit the grid structure.

Now, we define grid networks. In E^2, the rectilinear distance between any two points $u_1 = (u_{x1}, u_{y1})$, $u_2 = (u_{x2}, u_{y2})$ is $r(u_1, u_2) = |u_{x1} - u_{x2}| + |u_{y1} - u_{y2}|$. A rectilinear grid in E^2 consists of a set of parallel vertical lines and a set of parallel horizontal lines with a well-defined spacing. All the vertical lines (horizontal lines) are of the same length. Thus, a grid encloses a rectangular area in E^2. The intersection of a vertical line and a horizontal line defines an intersection point. Two adjacent intersections define a grid edge. Two adjacent horizontal (vertical) lines define a row (column). The intersection of a row and a column defines a cell. Figure 1.2 gives an example of a grid.

A grid must be treated as a network if the travel between any two points on the grid must be along the grid lines. This differentiates the grid network distances and the corresponding rectilinear distances. For any two points x and y, any shortest grid path connecting them travels in directions alternatively parallel to one or the other of the two axes in E^2. Points u_1 and u_2 are semi-antipodal to each other if they are both interior points of two different grid edges in the same row or column. The relation between the grid network distance $d(u_1, u_2)$ and the rectilinear distance $r(u_1, u_2)$ is

$$d(u_1, u_2) = r(u_1, u_2) + \delta(u_1, u_2),$$

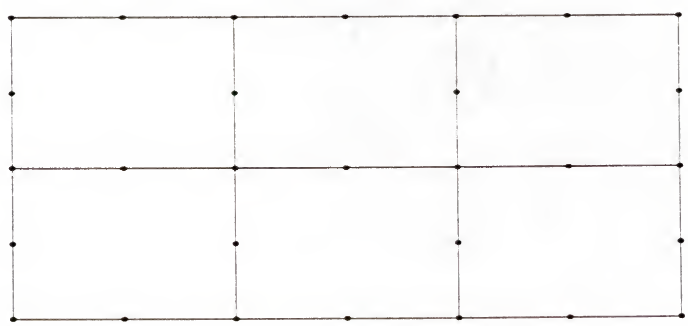


Figure 1.2 A Grid Network

where $\delta(u_1, u_2) \geq 0$ with $\delta(u_1, u_2) > 0$ if and only if points u_1 and u_2 are semi-antipodal to each other. Thus, if L^* denote the maximum of the grid edge lengths, then for any two points we have $r(u_1, u_2) \leq d(u_2, u_2) \leq r(u_1, u_2) + L^*$.

A network is a grid network if one can find a grid embedding in E^2 by adding finitely many artificial vertices as intersections in the grid. For example, a single cycle network can have a grid embedding in E^2 by adding at most three artificial vertices as intersections. We denote the grid embedding of a network as N_g. A vertex that is not an intersection is called an interior vertex. A grid network N_g may exist in reality (e.g. the road network of a city) in which case there are no artificial intersections. With N_g as a grid, all the definitions and claims about a grid are valid here. Grid N_g encloses a rectangular area in E^2, which we denote as N_r.

We now give the motivation for studying grid networks. Theoretically, a grid network is cyclic. Tamir (1993) showed that a special case of the grid network multimedian problem, when the network is a single cycle with three nodes, is strongly NP-hard. Yet, it is "close" to a rectilinear grid on which some multifacility problems can be solved. We can use the corresponding rectilinear problem as an approximation. On the other hand, the regularity of a grid network may enable us to do some analysis which is otherwise impossible on a general cyclic network. In applications, there are some cases, such as traveling in city streets, along the aisles of a workshop or a warehouse, or along an AGV guide path network, where the travel pattern is alternatively along the two axes of E^2. People generally use rectilinear distances to approximate the real distances. It is interesting to note that, to date, researchers seem to have just accepted rectilinear distance as a satisfactory approximation to grid network distance. The two distances are not the same, and the quality of the approximation has not been studied. We show that there are cases where the approximations are poor. This motivates studying location problems directly defined on grid networks. There is little study of grid network location problems as such, although we believe these to be an important, and reasonably tractable, class of cyclic network location problems.

In the following, we discuss literature related to grid networks. Larson and Sadiq (1983), and Batta et al. (1989) considered the p-median problem with rectilinear distances in the presence of barriers (areas which blocking travels) and convex forbidden regions (areas in which locating new facilities is not allowed). In Batta et al. (1989), they conclude that, with barriers and forbidden regions, the properties of the problem resemble the same kind of problem defined on a general cyclic network because of the loss of the gated subnetwork property (in their term, the single assignment property). Batta et al. (1989) and Batta and Chiu (1989) noticed that there are some relations between the rectilinear metric and the network metric. Bandelt (1989) studied the 1-median problem on a median network which, in fact, is a multidimensional rectilinear grid. Based on the gated subnetwork structure, he concluded that the set of 1-medians is a connected subnetwork and every local 1-median is also a global 1-median. Egbelu (1982), Tansel and Kiran (1988), Kiran and Tansel (1989), Goetz and Egbelu (1989) discussed problems of locating loading/unloading points in an AGV guide path network.

There is also some research in graph theory which is related to location problems on special cyclic networks (Proskurowski, 1980a,b; Hedetniemi et al., 1982, 1986; and Nieminen, 1988).

1.3 Overview

In Chapter 2, we consider the multimedian problem on a multiblock network. We obtain a localization result that localizes each new facility either to a block or a vertex. With the localization result, the problem can then be decomposed into smaller independent subproblems each of which is defined on a single block. These subproblems can be solved by branch and bound with the vertex-optimality property.

In Chapter 3, we study the multimedian problem on a grid network. We give some analytical results on the relations between the problem and a lower bounding problem -- the rectilinear multimedian problem. We also give a dominance relation for the multimedian problem on a grid network. Using this dominance relation, we find a polynomial-time algorithm that solves an approximation. We develop a branch and bound algorithm and give some computational results.

In Chapter 4, we study a general cyclic network multifacility problem with a convex function of distances. Based on some properties of distance functions, we propose using some piecewise linear and convex functions as their underestimates. We define a special type of solution subset which can be represented as a simple polytope in Euclidian space. For the subproblems on such a subset, we give various lower bounding techniques. We show that when the network is a grid network, we can make substantial improvement on the lower bounds, because the piecewise linear and convex underestimates of grid network distances are more useful and more available.

In Chapter 5, we give conclusion and future research remarks.

CHAPTER 2
THE MULTIMEDIAN PROBLEM ON A MULTIBLOCK NETWORK

In this chapter, we consider the multimedian problem P defined on a multiblock network G. From Chapter 1, a block of G can be an arbitrary cyclic network. Thus, the multimedian problem on a multiblock network is NP-Complete. Yet, by solving, in polynomial time, another multimedian problem on a tree -- a blocking graph, we can localize every new facility to either a vertex or a block. We then decompose the problem into independent multimedian problems, each of which corresponds to a localizing block.

This chapter is organized as follows. In Section 1, after introducing the necessary notation, we give our major localization result without proof and illustrate how to decompose the problem based on the localization. Section 2 discusses the insight for the localization result. Section 3 gives the proof of our main localization result. Because of the Vertex Optimality Property for the network multimedian problem, we only consider vertex solutions from now on in this chapter.

2.1 Localization and Decomposition

Localizing an optimal solution for a 1-median problem is considered by Goldman and Witzgall (1970), Goldman (1971), and Chen et al. (1985), with subsequent extensions by Love and Juel (1980) and Lefebvre et al. (1991) to versions of planar multimedian problems. Our result generalizes the result of Chen et al. to the multimedian problem.

First of all, we introduce the blocking graph. For a multiblock network G, its blocking graph BG is defined as follows. For every vertex v of G there is a vertex node cv in BG called the copy of v; for every block B in G, there is a block node CB in BG called the copy of B. An edge (cv, CB) is in BG if and only if $v \in B$. For convenience, the length of each edge in BG is one. It is known that BG is a tree if G is connected (Rosenstiel, Fiksel, and Holliger, 1972). Figure 2.1a gives a multiblock network with three blocks and two cut-points. Figure 2.1b is the

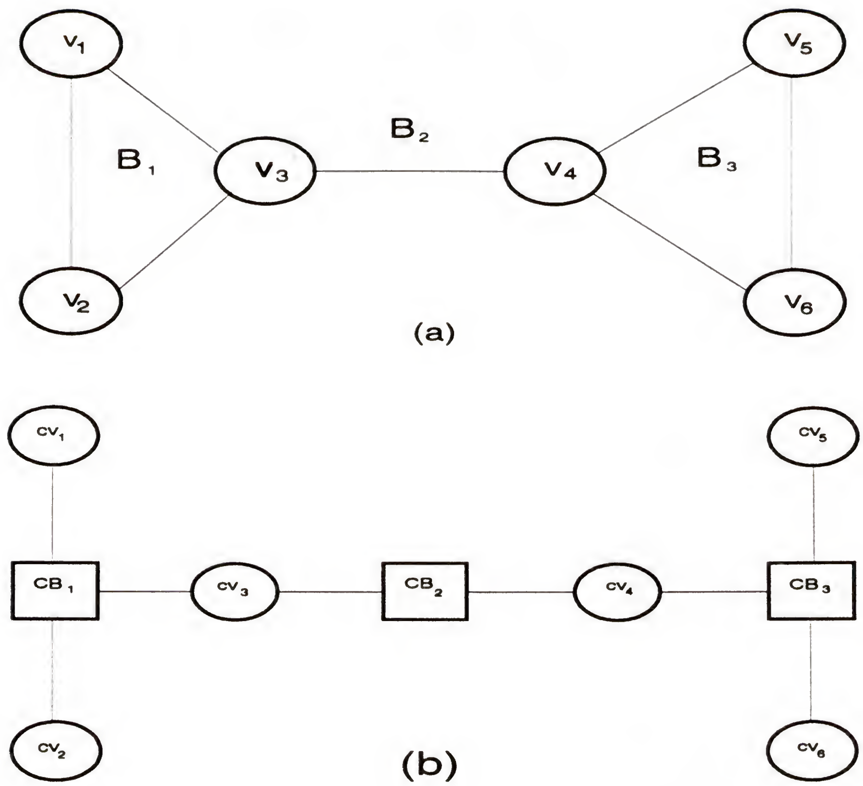


Figure 2.1 A Multi-Block Network and Its Blocking Graph

corresponding blocking graph. There is an $O(m^2)$ algorithm (Aho, Hopcroft, and Ullman, 1976) to construct the blocking graph of a network with m vertices.

For problem P, assigning every vertex node in BG the weights associated with the vertex in G and assigning zero weights to all the block nodes, we define a tree-multimedian problem on BG as follows.

$$\text{BP:}\ \underset{Z \in BG}{\text{Minimize}}\ F(Z) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij}\, d(cv_i, z_j) + \sum_{1 \le j < k \le n}\sum v_{jk}\, d(z_j, z_k).$$

Note that since the weights for the block-nodes are zero, they do not appear in the above expression. As with P, we are only interested in node solutions of BP.

Now we give our main localization result. Let Z be a solution to BP. In any given solution U to P, a new facility, say new facility j, is said to <u>conform</u> to Z if $u_j = v_s$ when $z_j = cv_s$ for any s, and $u_j \in B_q$ when $z_j = CB_q$ for any q. A solution U <u>conforms</u> to Z if every new facility in the solution conforms to Z.

Theorem 1. Let Z^* be an optimal solution to BP. There exists a vertex optimal solution U^* that conforms to Z^*.

There is an algorithm which solves a tree-multimedian problem in the order of $O(mn^3)$ (Kolen, 1982). Thus, if there are k blocks in a multiblock network, the localization conclusion needs $O((m+k)n^3)$ computation. We remark, since the localization conclusion of Theorem 1 is obtained by solving BP on the blocking graph, that no information about edge lengths of the network G is used. Likewise, the only information about each block that is used is which vertices are in the block, and which vertices of the block are cutpoints. Therefore, for any two multimedian problems with the same BG and the same induced block problem, the localization conclusions would be the same. In this sense, the localization information applies to a family of multimedian problems, and not just the specific problem of interest. Alternatively, we can view the blocking graph as being an aggregation of the original graph G, since it has a similar but simpler structure. Thus we can view BP as an approximation to P. Information we obtain about an optimal solution to BP will apply, in some sense, to P as well.

With Theorem 1, we now show how to decompose P. Without loss of generality, we assume that there is an optimal solution $Z^* = (cv_1, ..., cv_p, CB_{(p+1)}, ..., CB_{(n)})$. From Theorem 1, there exists an optimal solution to P with new facilities 1, ..., p vertex-localized to vertices $v_1, ..., v_p$ and new facilities p+1, ..., n block-localized to blocks $B_{(p+1)}, ..., B_{(n)}$. Decomposing P can be done in the following two steps.

(a) Removing Vertex-localized New Facilities

With new facility 1 localized to vertex v_1, the terms in the objective function of P involving new facility 1 are constant terms $w_{i1}d(v_i, v_1)$, i = 1, ..., m and variable terms $v_{1k}d(v_1, u_k)$, k = 2, ..., n. Each term $v_{1k}d(v_1, u_k)$ can be added to term $w_{1k}d(v_1, u_k)$ to create a new term $w_{1k}'d(v_1, u_k)$ where $w_{1k}' = w_{1k} + v_{1k}$, k = 2, ..., n. With $U\backslash\{1\} = \{u_2, ..., u_n\}$, $C_1 = \Sigma\{w_{i1}d(v_i, v_1) | i = 1, ..., m\}$, and $h(U\backslash\{1\})$ the objective function of the multimedian which only involves the last n−1 new facilities, we have

$$f(U) = h(U\backslash\{1\}) + C_1, \text{ for any U that conforms to } Z^*. \tag{2.1}$$

By removing all the vertex-localized new facilities one at a time in this way, we obtain

$$f(U) = h(u_{p+1}, ..., u_m) + C', \text{ for any U that conforms to } Z^*, \tag{2.2}$$

where C' is a constant. Thus, the multimedian problem equivalent to P is

P': Minimize $\{h(u_{p+1}, ..., u_m) \mid u_j \in B_{(j)}, j = p+1, ..., n\}$.

(b) Decomposing P'

Decomposing P' is based on the observation that cutpoint vertices serve the unique linkage for the "communication" between any two facilities in two different blocks. For a new facility j localized to a block B and a vertex v not in B, the unique closest point in B to v is a cutpoint vertex v' of B, such that

$$d(v, u_j) = d(u_j, v') + d(v', v). \tag{2.3}$$

For two new facilities j and k localized to two different blocks, say B_1 and B_2, there exists a unique cutpoint vertex v' in B_1 and a unique cutpoint vertex v" in B_2 such that

$$d(u_j, u_k) = d(u_j, v') + d(v', v'') + d(v'', u_k). \tag{2.4}$$

If B_1 and B_2 share a common vertex v, then v' = v" = v. Now, replace every term in h of P', which satisfies the conditions of equation (2.3) or (2.4) with the corresponding right hand side, and rearrange the terms by letting $f_{[i]}(U_{[i]})$ be the sum of weighted distances involving existing and new facilities in the same localizing block i, and letting C" be the sum of constant terms. We then have

$$h(u_{p+1}, ..., u_m) = \Sigma\{f_{[i]}(U_{[i]}) : i = p+1, ..., n\} + C''. \tag{2.5}$$

Combining (2.2) with (2.5), we have

$$f(U) = \Sigma\{f_{[i]}(U_{[i]}): i = p+1, ..., n\} + C' + C''. \tag{2.6}$$

Minimizing f(U) subject to U conforming to Z^* can then be done by minimizing each $f_{[i]}(U_{[i]})$ subject to $U_{[i]} \in B_{(i)}$, i = p+1, ..., n.

Example 2.1. For the network G shown in Figure 2.1a, let each edge length be 10 except for an edge length of 20 for edge (v_3, v_4). Define an instance, P_2, of the multimedian problem with 2 new facilities on this network with weights given in Table 2.1.

Table 2.1. Weight Data for Example 2.1.

w_{ij}	1	2	3	4	5	6
1	2	10	3	2	2	2
2	3	1	1	1	9	1

$v_{12} = 2$

Solution $Z^* = (CB_1, cv_5)$ is an optimal solution to BP_2 on the BG shown in Fig. 2.1b. From Theorem 1, there exists an optimal solution U^* to P_2 with $u_2^* = v_5$ and $u_1^* \in B_1$. With new facility 2 fixed at v_5, we can remove new facility 2 from further consideration by updating the weights of v_5; $w_{51}' = w_{51} + v_{12}$. Now $f(U) = h(u_1) + C_1$ where h( ) is the sum of all the terms of f(U) not involving new facility 2, $C_1 = \Sigma\{w_{i2}d(v_i, v_5): i = 1, ..., 6\} = 210$, and w_{15} is updated. Now with $u_1 \in B_1$, we have $d(v_i, u_1) = d(v_i, v_3) + d(v_3, u_1)$ for i = 4, 5, and 6. Thus,

$$h(u_1) = \sum_{i=1}^{3} w_{i1}d(v_i, u_1) + \sum_{i=4}^{6} w_{i1} [d(v_i, v_3) + d(v_3, u_1)].$$

Let $w_{11}' = w_{11}$, $w_{21}' = w_{21}$, $w_{31}' = [w_{31} + \sum_{i=4}^{6} w_{i1}]$, $C_2 = \sum_{i=4}^{6} w_{i1}\, d(v_i, v_3)$.

It is easy to verify that $h(u_1) = \sum_{i=1}^{3} w_{i1}'d(v_i, u_3) + C_2$. Thus, problem P_2 is reduced to solving a 1-median problem on block B_1.

It is still an open question as to how to solve a multimedian problem on a cyclic network which is a single block. There are two cases for which P can be easily solved.

<u>Case 1</u>. Many new facilities are vertex-localized, and each localizing block has few vertices.

<u>Case 2</u>. Each localizing block contains few localized new facilities.

2.2 The Tree-Like Structure

In this section, we will provide some insight for our localization result by drawing analogies from the tree multimedian problem. We also introduce necessary notation for the proofs of the main localization result in the next section.

Given any vertex v and any block B such that v ∈ B, network G will be separated into two connected components when one removes all the edges in B incident to v. We denote the component which contains vertex v as G(v, B) and the other as G(B, v). We call this pair the <u>gated pair (defined by v and B)</u> (gated pair for short) since vertex v is a gate vertex for both components, following Goldman and Witzgall (1970). Figure 2.2 gives an example of a gated pair defined by block B_3 and cut-point vertex v_4 in Figure 2.1a. For the trivial case where v is not a cutpoint vertex, G(v, B) = {v}.

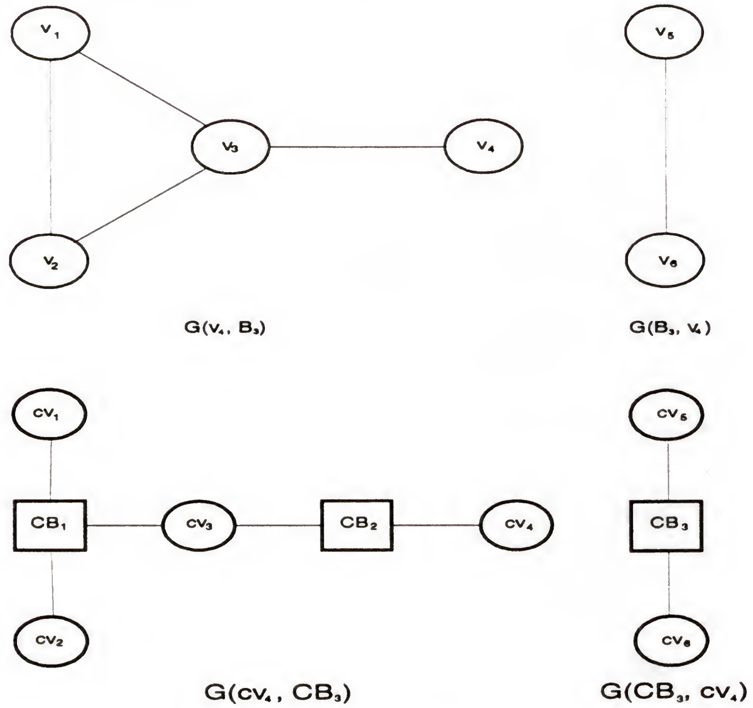


Figure 2.2 A Gated Pair and Its Copy

In the terminology of Goldman and Witzgall (1970), the two components G(v, B) and G(B, v) are gated subnetworks with gate point v, so that d(x, y) = d(x, v) + d(v, y), for any $x \in$ G(v, B) and $y \in$ G(B, v). Breaking down distance function d(x, y) involving variables x and y in the two components respectively into two distance functions d(x, v) and d(v, y) each only involves one variable in the respective locality, it indicates that the internal structure in G(B, v) (G(v, B)) has no effect on the objective function value when one only changes the new facility locations in G(v, B) (G(B, v)). This homogeneity effect on the distances is the basis for the results of the tree-multimedian problem (Goldman and Witzgall, 1970; Kolen, 1986) as well as for our localization result.

For a tree network, this homogeneity effect can be seen from the following two properties. For a tree network T, let (v_s, v_t) be an arbitrary edge and (T_s, T_t) be the subtree pair obtained by removing edge (v_s, v_t). For an instance of multimedian problem on T, a bi-partition (J_s^*, J_t^*) of the new facility indices over edge (v_s, v_t) is an optimal one, if there is an optimal solution with the new facilities in J_s^* (J_t^*) located in T_s (T_t).

Property 2.1. (Kolen, 1986) A bi-partition (J_s^*, J_t^*) over any given (v_s, v_t) is optimal if and only if, by initially locating all the new facilities on v_s, moving new facilities in J_t^* to v_t decreases the objective function value the most among all the choices of subsets of new facilities to move.

In the following, we give a necessary condition for an optimal bi-partition. This property tells when a movement of some new facilities to an adjacent vertex will or will not increase the objective function value. We will use this property in proving Theorem 1 in Section 3.

Property 2.2. Let (v_s, v_t) be an edge of a given tree T. Let (J_s^*, J_t^*) be an optimal bi-partition over edge (v_s, v_t) and U a solution with $u_j \in T_s$ ($u_j \in T_t$) if $j \in J_s^*$ $(j \in J_t^*)$. For any U' obtained from U by moving a subset of new facilities located on v_s (on v_t) to v_t (to v_s), $f(U') - f(U) \geq 0$.

Returning to the multimedian problem P on a multiblock network, the following property demonstrates partially the homogeneity effect at a given cutpoint vertex. This property is based on the triangle inequality and the definition of a gated pair.

Definition 2.1. For any given solution U to P and any gated pair (G(v, B), G(B, v)), define the corresponding partition of J over (G(v, B), G(B, v)) as (J_v, J_B) where $J_v = \{j : u_j \in G(v, B)\}$ and $J_B = \{j : u_j \in G(B, v)\}$.

Definition 2.2. For every new facility j, define the "weight" of a subnetwork G' w.r.t. a new facility j as $W^{(j)}(G') \equiv \sum \{w_{ij}: v_i \in G'\}$, and for any subset J' of J, define the "weight" of subset J' w.r.t. a new facility j as $V^{(j)}(J') \equiv \sum \{v_{jk} : k \in J'\}$ with $V^{(j)}(\emptyset) = 0$. For the single new facility case, we use W(G') and V(J') to denote the equivalent terms $W^{(1)}(G')$ and $V^{(1)}(J')$.

Property 2.3. Let U be any given solution to P and let (G(v, B), G(B, v)) be any given gated pair. Let U' be the solution obtained from U by moving a subset S of new facilities currently located in G(B, v) to vertex v. Then

$$f(U) - f(U') \geq \sum \{\Delta_j(S)\, d(v, u_j): j \in S\} + f_{NN}(U{:}S)$$

where for each $j \in S$, $\Delta_j(S) = [W^{(j)}(G(v, B)) + V^{(j)}(J_v)] - [W^{(j)}(G(B, v)) + V^{(j)}(J_B \backslash S)]$, and $f_{NN}(U{:}S) = \sum\sum \{v_{jk} d(u_j, u_k): j < k, j, k \in S\}$.

Proof. Since only the locations of new facilities in S are changed,

$$f(U)-f(U') = \sum_{j\in S}\{\sum\{w_{ij}[d(v_i, u_j) - d(v_i, v)] \mid v_i \in G(v, B)\} + \sum\{w_{ij}[d(v_i, u_j) - d(v_i, v)] \mid v_i \in G(B, v)\}$$
$$+ \sum\{v_{jk}[d(u_j, u_k) - d(v, u_k)] \mid k \in J_v\} + \sum\{v_{jk}[d(u_j, u_k) - d(v, u_k)] \mid k \in J_B \backslash S\} \}$$
$$+ \sum_{j<k, j,k\in S} v_{jk}\, d(u_j, u_k).$$

For each facility in G(v, B) (i.e. an existing facility in G(v, B) or a new facility in J_v), the corresponding difference of the distances (the term in the square brackets) is $d(u_j, v)$. Replace this difference by $d(u_j, v)$. Based on the triangle inequality, for each facility in G(B, v) the corresponding difference of distances is no less than $-d(v, u_j)$. Replace this difference by $-d(u_j, v)$. Taking out the common factor $d(u_j, v)$, the result follows immediately. ∎

2.3 Localization

In this section, we give the proof of our main localization result. The localization is based on studying the weight distributions in G through the blocking graph BG.

Since BP is a tree-multimedian problem, the discussion in Section 2.2 about the tree-multimedian problem goes through here. To relate the optimal information of BP to that of P, we have the following observations. As with an arbitrary tree network, edge (cv, CB) defines two subtrees. Let subtree T(cv, CB) be the one containing cv and T(CB, cv) be the other subtree. Then, T(cv, CB) is a "copy" of G(v, B) in the sense that every vertex in G(v, B) has its copy in T(cv, CB). Similarly, T(CB, cv) is a "copy" of G(B, v). Figure 2.2b gives the copies of the gated pair of Figure 2.2a in the corresponding blocking graph. Thus, an edge (cv, CB) in BG defines a gated pair (G(v, B),G(B, v)) in G. Consequently, we have

Remark 2.1. $W^{(j)}(G(v, B)) = W^{(j)}(T(cv, CB))$ and $W^{(j)}(G(B, v)) = W^{(j)}(T(CB, cv))$ for all j.

Using Kolen's Algorithm 2.1, one can determine an optimal bi-partition for BP over any given edge (cv, CB) of BG. In the following, we will show that any such an optimal bi-partition for BP determines an optimal bi-partition for P on the gated pair (G(v, B), G(B, v)). This optimal bi-partition property is represented in the form of dominating relations in the Lemma below.

For a given optimal solution Z^* to BP, let (J_{cv}^*, J_{CB}^*) be an optimal bi-partition for BP over an arbitrary edge (cv, CB) of BG. For a given solution U to P, let (J_v, J_B) be the bi-partition of J (see Definition 2.1) over the corresponding gated pair (G(v, B), G(B, v)). Let $L = J_{cv}^* \cap J_B$ and $R = J_{CB}^* \cap J_v$. Sets L and R represent the inconsistency between these two bi-partitions. To reduce inconsistency, we move new facilities in L to vertex v. Denoted by U^L the resulting solution, we will prove, in the Lemma, that U^L dominates U (i.e. $f(U) \geq f(U^L)$). First of all, we need the following two properties on the lower bounds for $f(U) - f(U^L)$.

Property 2.4. $f(U) - f(U^L) \geq \beta^0(L) = \sum_{j \in L} \delta_j(L)\, d(v, u_j) + f_{NN}(U : L)$, where for each $j \in L$

$$\delta_j(L) = [W^{(j)}(G(v, B)) + V^{(j)}(J_{cv}^* \setminus L)] - [W^{(j)}(G(B, v)) + V^{(j)}(J_{CB}^*)].$$

Proof. The movement of changing U to U^L here is the same as that defined in Property 2.3 with the set L to be the set S in Property 2.3. Thus, we have

$$f(U) - f(U^L) \geq \sum \{\Delta_j(L)\, d(v, u_j) \mid j \in L\} + f_{NN}(U : L)$$

where for each $j \in L$,

$$\Delta_j(L) = [W^{(j)}(G(v, B)) + V^{(j)}(J_v)] - [W^{(j)}(G(B, v)) + V^{(j)}(J_B \setminus L)].$$

Since $J_v = (J_{cv}^* \setminus L) \cup R$ and $J_B \setminus L = J_{CB}^* \setminus R$, we have $V^{(j)}(J_v) = V^{(j)}(J_{cv}^* \setminus L) + V^{(j)}(R)$ and $V^{(j)}(J_B \setminus L) = V^{(j)}(J_{CB}^*) - V^{(j)}(R)$. Thus,

$$\Delta_j(L) \geq [W^{(j)}(G(v, B)) + V^{(j)}(J_{cv}^* \setminus L)] - [W^{(j)}(G(B, v)) + V^{(j)}(J_{CB}^*)] = \delta_j(L).$$

Each $\Delta_j(L) \geq \delta_j(L)$ implies the result of this property. ∎

We see that the $f_{NN}(U : L)$ in this lower bound is still location dependent. The following property gives a lower bound which is location independent.

Property 2.5. By renumbering new facilities if necessary, assume that $L = \{1, ..., p\}$ for some p, $1 \leq p \leq n$, and that $d(v, u_h) \leq d(v, u_{h+1})$, $h = 1, ..., p-1$. Define

$$\theta_1 = d(v, u_1),\ \theta_h = d(v, u_h) - d(v, u_{h-1}) \text{ (Note that } \theta_h \geq 0),\ h = 2, ..., p;$$

Then $f(U) - f(U^L) \geq \beta(L)$ where $\beta(L) = \sum_{h=1}^{p} [\sum_{j=h}^{p} \delta_j(L \backslash \{1, ..., h-1\})] \theta_h$.

Proof. From Property 2.4, $f(U) - f(U^L) \geq \beta^0(L)$. Note that

$d(v, u_j) = \sum_{h=1}^{j} \theta_h$, $j = 1, ..., p$, $d(u_k, u_j) \geq d(v, u_j) - d(v, u_k) = \sum_{h=1}^{j}\theta_h - \sum_{h=1}^{k}\theta_h = \sum_{h=k+1}^{j} \theta_h$, $k < j$, $j, k \in L$.

Substituting these equalities and inequalities into $\beta^0(L)$ gives

$$\beta^0(L) \geq \sum_{j=1}^{p} [\delta_j(L) (\sum_{h=1}^{j} \theta_h)] + \sum_{k=1}^{p-1} [\sum_{j=k+1}^{p} v_{kj} (\sum_{h=k+1}^{j} \theta_h)] \equiv LB.$$

Changing the order of addition in LB and collecting the terms associated with the same θ_h,

$$LB = \sum_{j=1}^{p} \delta_j(L) \theta_1 + \sum_{h=2}^{p} \sum_{j=h}^{p} [\delta_j(L) + \sum_{k=1}^{h-1} v_{kj}] \theta_h.$$

Note that $\delta_j(L)+V^{(j)}(S) = \delta_j(L \backslash S)$ for any $S \subset L$, $j \notin S$. Thus, with $\sum_{k=1}^{h-1} v_{kj} = V^{(j)}(\{1, ..., h-1\})$, we have

$$LB = \sum_{j=1}^{p} \delta_j(L) \theta_1 + \sum_{h=2}^{p} \sum_{j=h}^{p} \delta_j(L \backslash \{1, ..., h-1\}) \theta_h = \beta(L).$$

The conclusion now follows. ∎

With the lower bound in Property 2.5, we now show that $f(U) - f(U^L) \geq 0$. We would like to point out the resemblance of this dominance relation to the Majority Theorem of Goldman and Witzgall (1970) for the 1-median problem.

Lemma. For any given solution U to P, for any given edge (cv, CB) in BG, and an optimal bi-partition for BP, let L, R, and U^L be the terms associated with solution U, the edge (cv, CB) and the optimal bi-partition. Then, $f(U) \geq f(U^L)$.

Proof. By re-numbering the new facilities if necessary, assume that $L = \{1, ..., p\}$ and $d(v, u_h) \leq d(v, u_{h+1})$, for $h = 1, ..., p-1$. Thus, all the conditions in Property 2.5 are satisfied. We only need to show that $\beta(L) \geq 0$

Denote by K_h the term $\sum_{k=h}^{p} \delta_j(L \backslash \{1, ..., h-1\})$ in $\beta(L)$. We have

$$K_h = \sum_{k=h}^{p} \delta_j(\{h, ..., p\})$$

$$= \sum_{j=h}^{p} \{[W^{(j)}(G(v, B)) + V^{(j)}(J_{cv}^{*}\backslash\{h,...,p\})] - [W^{(j)}(G(B, v)) + V^{(j)}(J_{CB}^{*})]\}$$

$$= \sum_{j=h}^{p} \{[W^{(j)}(T(cv, CB)) + V^{(j)}(J_{cv}^{*}\backslash\{h,...,p\})] - [W^{(j)}(T(CB, cv)) + V^{(j)}(J_{CB}^{*})]\}.$$

The last sum equals to F(Z') - F(Z) in problem BP, with Z, Z' the solutions of BP defined as follows. In Z each new facility j is located on cv (on CB) if $j \in J_{cv}^*$ ($j \in J_{CB}^*$) and Z' is obtained from Z by moving new facilities h, ..., p from cv along edge (cv, CB) into CB. Since (J_{cv}^*, J_{CB}^*) is an optimal bi-partition for BP on (cv, CB), this movement is the movement defined in Property 2.2. Thus, we have $F(Z') - F(Z) \geq 0$. This show that each $K_h \geq 0$. Since $\beta(L) = \sum K_h \theta_h$ and each $\theta_h \geq 0$, we have the result. ∎

The following corollary is the one that we think gives the insight for our localization result.

Corollary. Let (J_{cv}^*, J_{CB}^*) be an optimal bi-partition of BP over subtree pair (T_{cv}, T_{CB}) in BG for some vertex v and block B. Then, there exist an optimal solution to P such that all the new facilities in J_{cv}^* are located in G(v, B) and each new facility in J_{CB}^* is either located in G(B, v) or on vertex v.

Proof. We give the proof by showing that for any given solution U inconsistent with the location description of the corollary, we can find a solution, say U', that is consistent and $f(U') \leq f(U)$. Recall that U^L is the dominating solution for edge (cv, CB). Solution U^L is derived from U by moving new facilities in index subset J_{cv}^* but not currently located in G(v, B) to vertex v. From the Lemma, $f(U^L) \leq f(U)$. For this U^L, the inconsistency now comes from those new facilities in J_{CB}^* but currently located in those $G(B_{(k)}, v)$'s, where each $B_{(k)} \neq B$. Let the set of these new facilities be L'. Applying the Lemma to edges $(cv, CB_{(k)})$'s one at a time, we move the new facilities in L' to vertex v without increasing the objective value. The resulting solution is U'.

Now, we prove our main localization result. Recall that for any solution Z to BP and any solution U to P, a new facility j conforms to Z if $u_j = v_s$ when $z_j = cv_s$ for any s, and $u_j \in B_q$ when $z_j = CB_q$ for any q. Solution U conforms to Z if every new facility in U conforms to Z.

Theorem 1. Let Z^* be an optimal solution to BP. There exists a vertex optimal solution U^* that conforms to Z^*.

Proof. We will show that for any solution U, we can construct a solution U" conforming to Z^* with $f(U") \leq f(U)$.

If in U a new facility j does not conform to Z, then either $u_j \neq v_s$ when $z_j = cv_s$ for some s, or $u_j \notin B_q$ when $z_j = CB_q$ for some q. For the first case, new facility j must be in some $G(B_k, v_s)$ for some B_k. With gated pair $(G(v_s, B_k), G(B_k, v_s))$, the corresponding edge (cv_s, CB_k) in BG, and solution U, we can obtain a dominating solution U^L by applying the Lemma to the edge (cv_s, CB_k). In deriving U^L from U, a set of non-conforming new facilities in $G(B_k, v)$, including new facility j, are moved to v_s without increasing the objective function value. For the second case, new facility j must be in some G(B', v') for some cutpoint vertex v' connecting block B_q and B'. Similar to the first case, we can move a set of non-conforming new facilities, including new facility j, to v' without increasing the objective function value. In both cases, the resulting solution after the movement has at least one more new facility that becomes conforming since the movement does not move any conforming new facility. Thus, by performing at most n movements, we can construct the solution U". ∎

CHAPTER 3
THE MULTIMEDIAN PROBLEM ON A GRID NETWORK

In this chapter, we study the multimedian problem P_g on a grid network N_g, where

$$P_g: \underset{U \in N_g^n}{\text{Minimize}}\ f(U) = \sum_{j=1}^{n} f_j(u_j) + f_{NN}(U),$$

with $f_j(u_j) = \sum_{j=1}^{m} w_{ij}\, d(v_i, u_j)$, $j = 1, \ldots, n$, and $f_{NN}(U) = \sum_{1 \leq j < k \leq n} v_{jk}\, d(u_j, u_k)$.

In some applications of the multimedian problem, such as the facility layout, locating pickup/loading point, and locating warehouses on a city street network, the networks encountered are often grids or grid-like. Though grid networks are specialized cyclic networks, problem P_g is strongly NP-hard (Tamir 1993). Yet, using the rectilinear distance underestimates of grid network distances, we construct a polynomial-time solvable rectilinear multimedian problem P_r as a lower bounding problem for P_g. Problem P_r is asymptotic to P_g as the grid network becomes "closer" to a rectilinear grid. While it has been widely assumed that this approximation relationship between the two types of distances is satisfactory, we know of no studies conducted, either theoretically or experimentally, about this approximation. This chapter is, then, to study this approximation with respect to the multimedian problem in order to solve P_g better.

This chapter is organized as follows. Section 3.1 studies the relationship between P_r and P_g and gives a dominance relation for P_g. Section 3.2 considers finding a near-optimal solution to P_g based on an optimal solution to P_r. Due to the dominance relation in Section 3.1, we find a polynomial-time algorithm to solve problem P_g^I – a subproblem of P_g, in which new facilities are restricted to grid intersections. Our computational experience suggests that P_g^I is, on average, a good approximation of P_g. Section 3.3 proposes a branch and bound scheme. Section 3.4 reports computational results for the branch and bound algorithm.

3.1 Relations Between P_g and P_r

In this section we will give some theoretical results on the approximation relationships between P_r and P_g. We divide this section into four subsections. The first subsection reviews the decomposition of P_r. The second subsection discusses some basic relations between P_r and P_g. The third subsection discusses a useful dominance relation for P_g based on a given optimal solution of P_r. The final subsection is a worst-case analysis of the approximation.

3.1.1. Decomposition of P_r

A grid network has an embedding in the 2 dimensional Euclidean space E^2, so that every vertex on the grid network has coordinates in E^2. Let (v_{xi}, v_{yi}) denote the coordinates of vertex v_i and let (u_{xj}, u_{yj}) denote the coordinates of location variable u_j. Recall that N_r denotes the rectangular area enclosed by N_g. Rectilinear problem P_r can be expressed as:

$$P_r: \underset{Z \in N_r^n}{\text{Minimize}}\ h(Z) = \sum_{j=1}^{n} h_j(z_j) + h_{NN}(Z),$$

where the h_j's and h_{NN} are obtained by replacing the grid network distances in the f_j's and f_{NN} by the corresponding rectilinear distances. It is well-know that P_r can be decomposed into two independent multimedian problems (Francis, McGinnis, and White, 1992) as follows:

$$P_{rx}: \underset{Z_x \in R^n}{\text{Minimize}}\ h_x(Z_x) = \Sigma_j \Sigma_i w_{ij}|v_{xi} - z_{xj}| + \Sigma_{1\le j<k\le n} v_{jk}|z_{xj} - z_{xk}|, \text{ and}$$

$$P_{ry}: \underset{Z_y \in R^n}{\text{Minimize}}\ h_y(Z_y) = \Sigma_j \Sigma_i w_{ij}|v_{yi} - z_{yj}| + \Sigma_{1\le j<k\le n} v_{jk}|z_{yj} - z_{yk}|.$$

Problems P_{rx} can be transformed into multimedian problems on path networks $T_x = (V_x, E_x)$ as follows. Let $I_x = \{s_{x1}, s_{x2}, ..., s_{xp}| s_{xi} < s_{xi+1}, i = 1, ..., p-1\}$ be the set of the distinct x-coordinates of the vertices of N_g. Then, T_x is the path network of p nodes $t_{x1}, ..., t_{xp}$ with t_{xi} adjacent to t_{xi+1} and $d(t_{xi}, t_{xi+1}) = s_{xi+1} - s_{xi}$, $i = 1, ..., p-1$. For each vertex t_{xi}, assign weights $w_{ij}^x = \Sigma\{w_{hj} \mid v_{xh} = s_{xi}\}$, $i = 1, ..., p$, $j = 1, ..., n$. Problem P_{rx} thus can then be expressed as

$$P_{rx}: \underset{Z_x \in T_x^n}{\text{Minimize}}\ \Sigma_j \Sigma_i w_{ij}^x d(t_{xi}, z_{xj}) + \Sigma_{1\le j<k\le n} v_{jk} d(z_{xj}, z_{xk})$$

With path network T_y similarly constructed and weights on T_y similarly assigned, problem P_{ry} can be expressed as

P_{ry}: $\underset{Z_y \in T_y^n}{\text{Minimize}} \sum_j \sum_i w_{ij}{}^y d(t_{yi}, z_{yj}) + \sum_{1 \le j < k \le n} v_{jk} d(z_{yj}, z_{yk})$.

It is well-know that a solution Z^* is optimal to P_r if and only if Z_x^* and Z_y^* are optimal to P_{rx} and P_{ry} respectively.

3.1.2. Some Basic Relations between P_g and P_r

With the embedding of N_g in E^2, a solution U to P_g is also a vector of n points in E^2, so that U is a solution to P_r. Since rectilinear distances are underestimates for the corresponding grid network distances, we know the following.

Remark 3.1. For any solution U to P_g, $h(U) \le f(U)$, with equality holding if and only if

a. $w_{ij}d(v_i, u_j) = w_{ij}r(v_i, u_j)$, $\forall i, j$, and b. $v_{jk}d(u_j, u_k) = v_{jk}r(u_j, u_k)$, $\forall j, k$.

One special case for which conditions a and b are true is when U is an intersection solution. Conditions a and b can serve as a measure of the approximation a Z^* has. Let a closest solution U' to Z^* be a solution with each u_j' one of the closest points on N_g to z_j^*. If U' has few violations in a and b and h(U') is "close" to $h(Z^*)$, we would consider Z^* a good approximate solution of P_g. One extreme case is when Z^* is a solution to P_g and it satisfies conditions a and b above. In this case, Z^* is an optimal solution to P_g.

3.1.3. A Dominance Relation

Since the solution set for P_g is contained in the solution set for P_r, we can use some necessary optimality conditions for P_r to obtain some useful dominance relations for P_g. In this subsection, we describe one.

3.1.3.1. Some A-Posteriori Dominance Relations for P_r

As far as we know, the only known a-posteriori dominance relation for MMP is related to the convexity property of MMP defined on tree networks. That is, for an optimal solution X^* and another solution X of some MMP on a tree network, the set $\{Z \mid Z = \lambda X + (1-\lambda)X^*, 0 \le \lambda \le 1\}$ is a dominant solution subset of X (Dearing et al. 1976). Unfortunately, this specific definition of convex combinations is too restrictive to help in our subsequent analysis. In the following, we introduce a similar concept, and use this concept to express an a-posteriori dominance relation for MMP defined on path networks. This dominance relation on path networks is then utilized to obtain an a-posteriori dominance relation for P_r.

Definition 3.1. For any given n-vectors X, Y on a given tree network T, a λ-combination $\lambda X + (1-\lambda)Y$, $\lambda = (\lambda_1, ..., \lambda_n)^T$, $0 \le \lambda_j \le 1$, is an n-vector Z with z_j the point a distance of $(1-\lambda_j)d(x_j, y_j)$ from x_j on the path connecting x_j and y_j, j = 1, ..., n.

The difference between the λ-combination and the convex combination is in that the latter requires that any two λ_j, λ_k are identical. Now, we concentrate on path networks. Let T be a path network with one end point designated as the origin, so that T is an ordered set. Thus, for any two n-vectors $X = (x_1, ..., x_n)$ and $Y = (y_1, ..., y_n)$ of points on T, we have either $x_j < y_j$, $x_j = y_j$, or $x_j > y_j$ for each j. Consider the following definition of a partition of the index set {1, ..., n} based on the relative positions of x_j's about the corresponding y_j's.

Definition 3.2. For any given X, Y ∈ T^n, let $L(X \mid Y) = \{j \mid x_j \le y_j\}$ and $R(X \mid Y) = \{j \mid y_j < x_j\}$.

For any X and Y ∈ T^n, let Z be a λ-combination of X and Y. Then, for each z_j we either have $x_j \le z_j \le y_j$ if $x_j \le y_j$, or $y_j \le z_j \le x_j$ if otherwise.

Definition 3.3. For a λ-combination Z of X, Y ∈ T^n, let $L(X \mid Y)^+ = \{k \mid x_k < z_k\}$ and $R(X \mid Y)^+ = \{k \mid z_k < x_k\}$. Let $L(X \mid Y)^0 = L(X \mid Y) - L(X \mid Y)^+$ and $R(X \mid Y)^0 = R(X \mid Y) - R(X \mid Y)^+$ (It is easy to see that $L(X|Y)^+ \subseteq L(X \mid Y)$ and $R(X|Y)^+ \subseteq R(X \mid Y)$).

Definition 3.4. A λ-combination Z of X and Y is said to be ordered like Y if it satisfies the following two conditions:

a. for any $k \in L(X \mid Y)^+$ and any $j \in L(X \mid Y)$, if $z_j < z_k$ then $y_j < y_k$;

b. for any $k \in R(X \mid Y)^+$ and any $j \in R(X \mid Y)$, if $z_j > z_k$ then $y_j > y_k$;

What Definition 3.4 says is that Z is ordered like Y, if the order relation between z_j and z_k is the same as that between y_j and y_k for every index pair (j, k) in $S(Z) = \{(j, k) | k \in L(X \mid Y)^+, j \in L(X \mid Y), z_j < z_k\} \cup \{(j, k) | k \in R(X \mid Y)^+, j \in R(X \mid Y), z_j > z_k\}$. This order conformity is similar to a more common one (call it conformity C) - two vectors Y and Z have order conformity

C if and only if there exists a mapping σ such that $z_{\sigma(1)} < z_{\sigma(2)} < \ldots < z_{\sigma(n)}$ and $y_{\sigma(1)} < y_{\sigma(2)} < \ldots < y_{\sigma(n)}$; or equivalently that for every index pair (j, k), if $z_j < z_k$ then $y_j < y_k$. Definition 3.4 requires order conformity for a <u>subset</u> S(Z) of index pairs, whereas order conformity C requires order relation conformity for every index pair. Hence, the latter one implies the former.

<u>Example 3.1</u>. For the X and Y in Figure 3.1, we have $L(X \mid Y) = \{2, 3\}$ and $R(X \mid Y) = \{1\}$. The Z's shown in Figures 3.1a, ..., 3.1d are all λ-combinations of X and Y. Since $R(X \mid Y)$ is a singleton, $R(X \mid Y)^+ \subseteq R(X \mid Y)$ will be either an empty set or a singleton. Therefore, there exist no index pairs satisfying the premise of condition b, so that any λ-combination of X and Y satisfies condition b vacuously. Therefore, any λ-combination of X and Y satisfying Condition a of Definition 3.4 is ordered like Y.

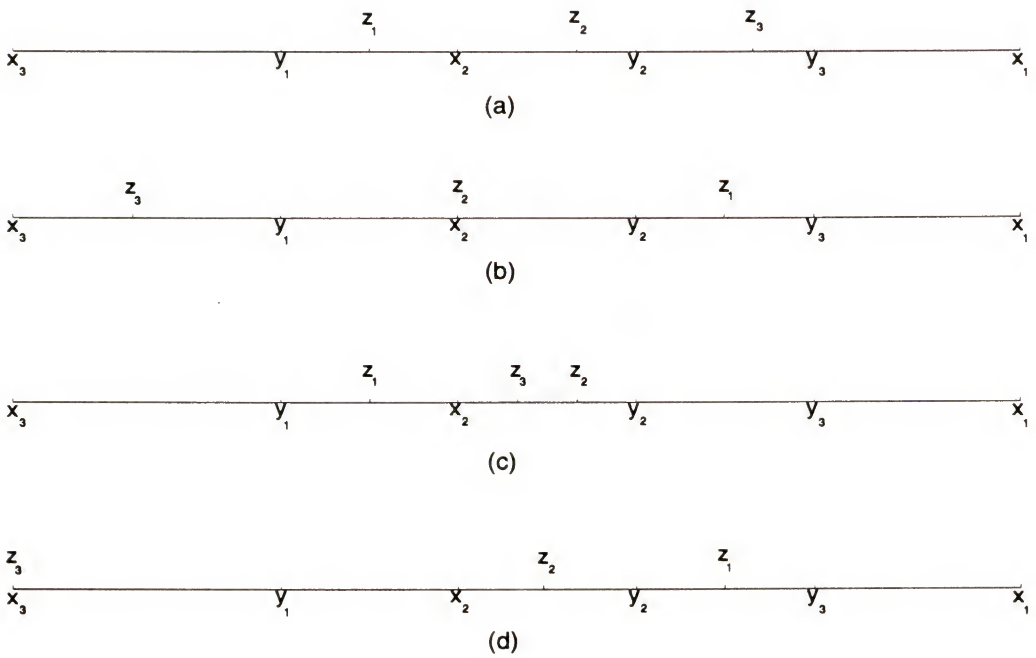


Figure 3.1 λ-Combinations with Z ordered like Y in (a) and (b) only

For the Z shown in Figure 3.1a, we have $L(X \mid Y)^+ = \{2, 3\}$ and $R(X \mid Y)^+ = \{1\}$. Since both $z_2 < z_3$ and $y_2 < y_3$, Condition a is satisfied. Therefore, Z is ordered like Y. For the Z shown in Figure 3.1b, we have $L(X \mid Y)^+ = \{3\}$. In this case, Condition a is vacuously true since there are no index pairs satisfying the premise of Condition a. For the Z shown in Figure 3.1c, we have $L(X \mid Y)^+ = \{2, 3\}$. Since $z_3 < z_2$ but $y_3 > y_2$, Condition a is not satisfied, so that Z is not ordered like Y. Finally, for the Z shown in Figure 3.1d, we have $L(X \mid Y)^+ = \{2\}$. Again, since $z_3 < z_2$ but $y_3 > y_2$, Z is not ordered like Y.

As a final note, we see that a convex combination defined by Dearing et al. is a special case of a λ-combination. A convex combination is not necessarily ordered and a λ-combination ordered is not necessarily a convex combination. Now, we give an a-posteriori dominance relation for MMP on path networks. The proof is in Appendix A.

Property 3.1. Let P be a MMP on a path network T. For an optimal solution X^* and another arbitrary solution X to P, any λ-combination X' of X and X^* ordered like X^* dominates X (i.e. $f(X') \leq f(X)$, where f is the objective function of P).

Since problem P_r can be decomposed into two independent MMP's, P_{xr} and P_{yr}, on path networks, Property 3.1 can be used to obtain an a-posteriori dominance relation for P_r.

Property 3.2. Let Z^* be an optimal solution to P_r and Z be any other solution to P_r. Then, a solution Z' dominates Z, if

a. Z_x' is a λ-combination of Z_x and Z_x^* and is ordered like Z_x^*, and

b. Z_y' is a λ-combination of Z_y and Z_y^* and is ordered like Z_y^*.

Proof. We know that Z_x^* and Z_y^* are optimal solutions to P_{xr} and P_{yr} respectively. Thus, from condition a and Property 3.1, we have $h_x(Z_x') \leq h_x(Z_x)$. From condition b and Property 3.1, we have $h_y(Z_y') \leq h_y(Z_y)$. Hence, $h(Z') = h_x(Z_x') + h_y(Z_y') \leq h_x(Z_x) + h_y(Z_y) = h(Z)$. ∎

Based on Property 3.2, we will give a special case of the dominance relation for P_r, which is expressed in the geometric terms of N_g. This special case will then be used to develop some insightful dominance relations for P_g in the next subsection. First, we need to introduce some geometric terminology for N_g. Let $vl_{x1}, \ldots, vl_{xp}$ be the x-coordinates of the vertical grid lines and let $hl_{y1}, \ldots, hl_{yq}$ be the y-coordinates of the horizontal grid lines. For a Z^*, define lz_j^* and rz_j^* to be, respectively, the x-coordinate of the "left" and the "right" adjacent vertical grid lines of z_j^*; that is, $lz_j^* = \text{maximum}\{vl_{xi} \mid vl_{xi} \le z_{xj}^*\}$ and $rz_j^* = \text{minimum}\{vl_{xi} \mid vl_{xi} \ge z_{xj}^*\}$. Let bz_j^* and and tz_j^* be defined similarly for the bottom and top adjacent horizontal grid lines.

Definition 3.5. (Covering Row and Column) The covering column (covering row) of z_j^* is the set of points p such that $lz_j^* < p_x < rz_j^*$ ($bz_j^* < p_y < tz_j^*$).

In words, the covering column (covering row) of z_j^* is the set of interior points of the grid column (the grid row) in which z_j^* is an interior point. If z_j^* is an interior point of a column (a row) only, then the covering row (column) of z_j^* is an empty set. Therefore, if z_j^* is an intersection, then both the covering column and the covering row are empty sets.

Example 3.2. Figure 3.2 is an example illustrating the covering columns and rows for a given Z^*. The covering row of z_2^* is an empty set, since $bz_2^* = tz_2^*$. Similarly, the covering column and row of z_3^* are empty sets.

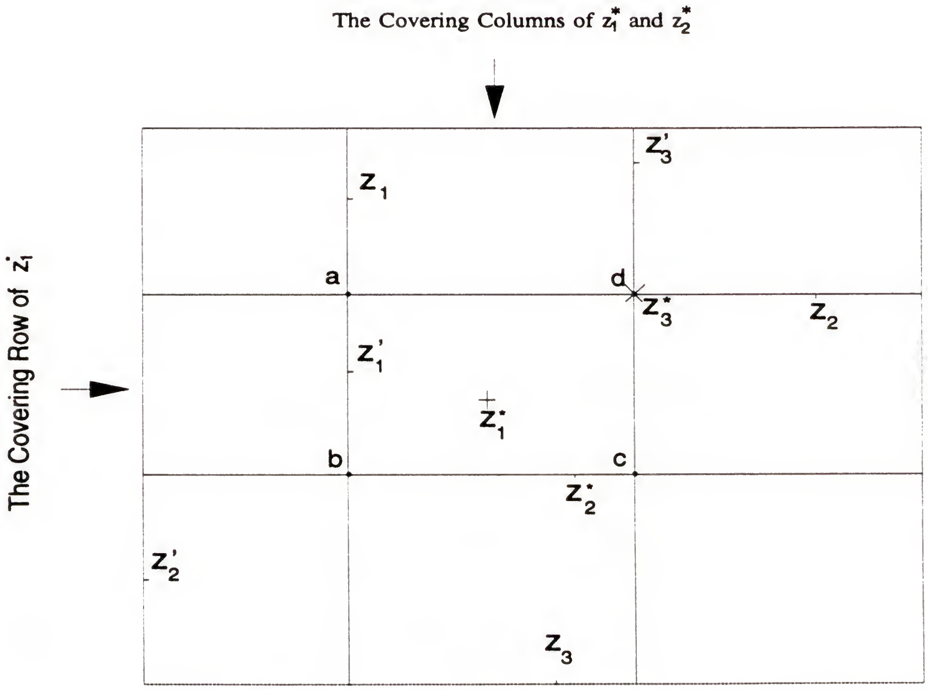


Figure 3.2 Covered and Uncovered Solutions

Definition 3.6. For any solution Z of P_r and an optimal solution Z^* of P_r, new facility location z_j is said to be covered if z_j is an interior point of the covering row and/or covering column of z_j^*. Solution Z is uncovered if every z_j is uncovered. Otherwise, when some z_j's are covered, we call solution Z covered.

For example, in Figure 3.2, solution Z is an uncovered solution and solution Z' is covered.

Definition 3.7. The set of neighboring intersection points of z_j^* is $NIP_j \equiv \{(lz_j^*, bz_j^*), (lz_j^*, tz_j^*), (rz_j^*, bz_j^*), (rz_j^*, tz_j^*)\}$. The neighbourhood rectangle N_j of z_j^* is the convex hull of the neighboring intersection points of z_j^*. A solution Z is a neighboring solution (NS) if $z_j \in N_j$, $j = 1, \ldots, n$. Otherwise, Z is a nonneighboring solution. A solution Z^I of P_r is a neighboring intersection solution (NIS) if $z_j^I \in NIP_j$, $j = 1, \ldots, n$.

As an example, for the Z^* given in Figure 3.2, we have $NIP_1 = \{a, b, c, d\}$, $N_1 = \{ p \in N_r | a_x \le p_x \le c_x, b_y \le p_y \le d_y\}$, $NIP_2 = \{b, c\}$, $N_2 = \{p \in N_r \mid b_x \le p_x \le c_x, p_y = b_y\}$, and $NIP_3 = N_3 = \{z_3^*\}$. We see that N_1 is a non-degenerate rectangle in N_r while N_2 and N_3 are degenerate.

Definition 3.8. (Closest Neighboring Solution)

Let Z^* be an optimal solution to P_r. For any nonneighboring solution Z of P_r, define its unique closest neighboring solution Z^c as the following. If $z_j \in N_j$ then $z_j^c = z_j$. If $z_j \notin N_j$, then let z_j^c be the unique closest point to z_j, in terms of the rectilinear distance, in N_j.

Remark 3.2. If a z_j is uncovered, then its closest point in N_j is an intersection point. Thus, for any uncovered solution Z of P_r, its closest neighboring solution is a neighboring intersection solution. If a z_j is covered, then there are three cases:

a. z_j is covered by both the covering column and covering row of z_j^*;

b. z_j is only covered by the covering column of z_j^*; and

c. z_j is only covered by the covering row of z_j^*.

For case a, since $z_j \in N_j$, we have $z_j^c = z_j$. For case b, z_j may or may not be in N_j. Nevertheless, we have

$$z_{xj}^c = z_{xj} \text{ and } z_{yj}^c = \begin{cases} tz_j^* \text{ if } z_{yj} \ge tz_j^* \\ bz_j^* \text{ if } z_{yj} \le bz_j^*. \end{cases}$$

Similarly, for case c we have

$$z_{yj}{}^{c} = z_{yj} \text{ and } z_{xj}{}^{c} = \begin{cases} rz_j{}^* \text{ if } z_{xj} \geq rz_j{}^* \\ lz_j{}^* \text{ if } z_{xj} \leq lz_j{}^*. \end{cases}$$

We see that for any j, the $z_{xj}{}^c$ is either $lz_j{}^*$, $rz_j{}^*$, or $z_{xj}{}^*$ depending on whether z_{xj} is to the left, to the right, or an interior point of grid interval $[lz_j{}^*, rz_j{}^*]$. Similarly, $z_{yj}{}^c$ is either $bz_j{}^*$, $tz_j{}^*$, or $z_{yj}{}^*$ depending on whether z_{yj} is to the left, to the right, or an interior point of grid interval $[bz_j{}^*, tz_j{}^*]$.

As an example, for the Z^* given in Figure 3.3, Z^c and $Z^{c'}$ are respectively the closest neighboring solutions of Z and Z'. We see that Z^c is a neighboring intersection solution. Since z_1' and z_2' are covered, we have $z_{x1}{}^{c'} = z_{x1}'$, $z_{y1}{}^c = bb_{y1}$, and $z_{y2}{}^{c'} = z_{y2}'$, $z_{x2}{}^{c'} = rb_{x2}$.

Now, we can give the a-posteriori dominance relation for P_r in the geometric terms of N_g.

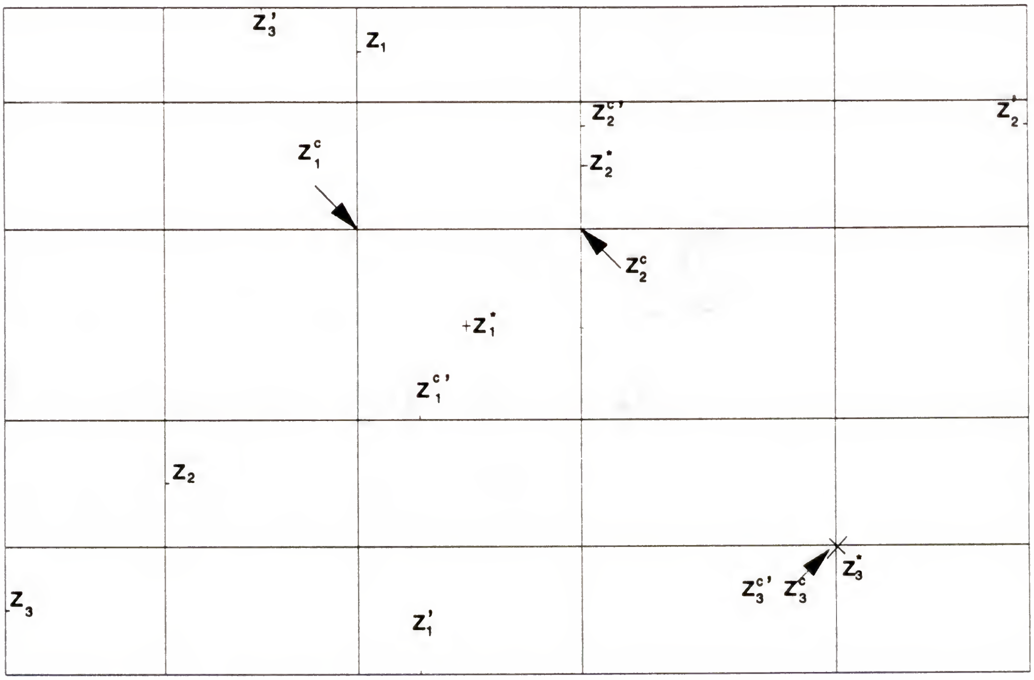


Figure 3.3 The Closest Neighboring Solutions

Property 3.3. Let Z^* be an optimal solution of P_r. Then, for any nonneighbor solution Z of P_r its closest neighboring solution Z^c dominates Z (i.e. $h(Z^c) \leq h(Z)$).

Proof. From Property 3.2, it is sufficient to show that Z_x^c is a λ-combination of Z_x and Z_x^* ordered like Z_x^*, and Z_y^c is a λ-combination of Z_y and Z_y^* ordered like Z_y^*. We only need to prove the first case, since, except for notation, the proof of the second case is the same.

Define a partition of the new facility index set J as $L^+ = \{j \mid z_{xj} < lz_j^*\}$, $R^+ = \{j \mid z_{xj} > rz_j^*\}$, $L^0 = \{j \mid lz_j^* \le z_{xj} \le z_{xj}^*\}$, and $R^0 = \{j \mid z_{xj}^* < z_{xj} \le rz_j^*\}$. From the definition of Z^c, we know that

$$z_{xk}^* \ge z_{xk}^c = lz_k^* > z_{xk}, \text{ for any } k \in L^+ \tag{3.1}$$

$$z_{xj}^* \le z_{xj}^c = rz_j^* < z_{xj}, \text{ for any } j \in R^+ \tag{3.2}$$

$$z_{xj}^* \ge z_{xj}^c = z_{xj}, \text{ for any } j \in L^0, \text{ and} \tag{3.3}$$

$$z_{xj}^* < z_{xj}^c = z_{xj} \text{ for any } j \in R^0, \tag{3.4}$$

From (3.1), ..., (3.4), z_{xk}^c is a point in the path connecting z_{xk} and z_{xk}^*, for k = 1, ..., n. This shows that solution Z^c is a λ-combination of Z_x and Z_x^*.

Now, we show that Z_x^c is ordered like Z_x^*. Let $L(Z_x \mid Z_x^*)$ and $R(Z_x \mid Z_x^*)$ be the sets of indices defined in Definition 3.2. From (3.1) to (3.4), sets L^+, L^0, R^+, and R^0 are the same sets defined in Definition 3.3. Therefore, we can use these sets to examine whether conditions a and b in Definition 3.4 are satisfied. Since Conditions a and b are symmetric, we only need to show that Condition a is satisfied. That is, for any $k \in L^+$ and any $j \in L(Z_x \mid Z_x^*)$, if $z_{xj}^c < z_{xk}^c$ then $z_{xj}^* < z_{xk}^*$. To prove by contradiction, suppose that there is a $j \in L(Z_x \mid Z_x^*)$ such that $z_{xj}^c < z_{xk}^c$ but $z_{xj}^* \ge z_{xk}^*$. This assumption implies that $lz_j^* \ge lz_k^*$, from the definitions of lz_j^* and lz_k^*. From (3.1), $z_{xk}^c = lz_k^*$, so that $z_{xj}^c < lz_k^* \le lz_j^*$. Since $j \in L(Z_x \mid Z_x^*)$, we know that $z_{xj} \le\le z_{xj}^c \le z_{xj}^*$, so that $z_{xj} < lz_j^*$. From the definition of z_{xj}^c, $z_{xj} < lz_j^*$ implies that $z_{xj}^c = lz_j^*$. Thus, $z_{xj}^c \ge lz_k^*$, so that $z_{xk}^c \le z_{xj}^c$ which contradicts to the assumption $z_{xj}^c < z_{xk}^c$. ∎

3.1.3.2. A Dominance Relation for P_g

Here, we will argue that for a given optimal solution Z^* to P_r, the neighborhood $(N_1 \times ... \times N_n) \cap N_g$ contains an optimal solution or a near-optimal solution to P_g. For any nonneighboring solution U of P_g and its closest neighboring solution U^c, we have

$$f(U) - f(U^c) = h(U) + \Delta(U) - (h(U^c) + \Delta(U^c))$$
$$= h(U) - h(U^c) + (\Delta(U) - \Delta(U^c)),$$

where $\Delta(U) = f(U) - h(U)$. From Property 3.2 we know that $h(U) - h(U^c) \geq 0$. We can show that

$$\Delta(U) - \Delta(U^c) = \Sigma_{j\in C'}\Sigma_{i\in A(j)}w_{ij}\delta(v_i, u_j) + \Sigma_{j\in C}[\Sigma_{i\in D(j)}w_{ij}\delta(v_i, u_j) - \Sigma_{i\in O(j)}w_{ij}\delta(v_i, u_j)],$$

here $\delta(.,.) = d(.,.) - r(.,.)$, C' and C are, respectively, the sets of uncovered and covered new facilities in U; A(j) is the set of vertices each of which has its grid network distance to u_j greater than the corresponding rectilinear distance; and D(j) (O(j)) is the set of vertices in the grid edge containing u_j^c (u_j). Value $\Delta(U) - \Delta(U^c)$ is likely to be non-negative, let alone that $h(U) - h(U^c)$ is usually larger than $|\Delta(U) - \Delta(U^c)|$. From the convexity nature of function h, the "further away" U is from U^c (in terms of $\Sigma_j r(u_j, u_j^c)$), the larger $h(U) - h(U^c)$. By contrast, $\Delta(U) - \Delta(U^c)$ is invariant to the "distance" from U to U^c. Thus, the "further away" U is from U^c, the more likely it is that $f(U) \geq f(U^c)$. When U is "near" U^c, we should have f(U) close to $f(U^c)$, so that if U is a near-optimal solution then U^c should also be a near-optimal solution. All these analyses indicate that <u>there often exists a near-optimal solution of P_g among the neighboring solutions of Z^*</u>.

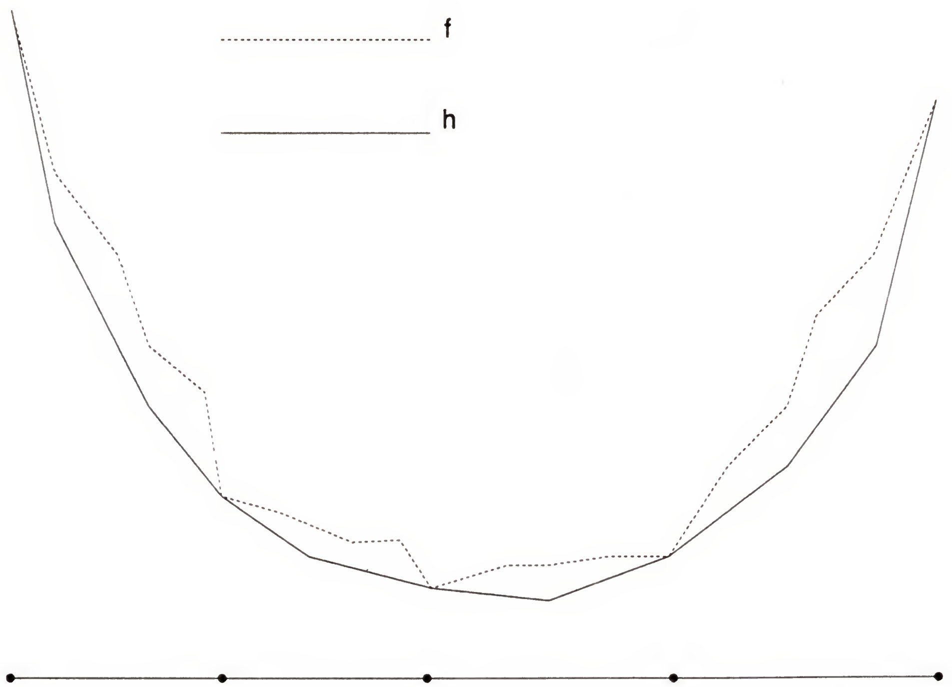


Figure 3.4 The Graphs of f and h

To help get insight into the convexity nature of h and the relative magnitude of δ compared with h, we include, in Figure 3.4, a <u>conceptual</u> illustration of the graphs of h(u) and f(u) of a single facility MMP, as u moves along a grid line.

Due to the variety of grid networks and the arbitrariness of the weight distribution pattern over the network, it is difficult to obtain general analytical results for the relationship between optimal solutions of P_r and optimal solutions of P_g. For example, though in most cases the neighbourhood $(N_1 \times \ldots \times N_n) \cap N_g$ of Z^* contains an optimal solution of P_g, there exist some extreme instances with optimal solutions not in the neighbourhood. Still, we are able to identify analytically a dominated set which is a subset of nonneighbor solutions of P_g.

<u>Corollary 3.3.1</u>. The set of neighboring intersection solutions of P_g dominates the set of uncovered solutions of P_g.

<u>Proof</u>. From Remark 3.2, the closest neighboring solution U^c of an uncovered U is a neighboring intersection solution. Thus, $\Delta(U^c) = 0$. Hence, $f(U) - f(U^c) = h(U) - h(U^c) + \Delta(U)$, where $\Delta(U) = f(U) - h(U)$ is a nonnegative term. From Property 3.3, $h(U) \geq h(U^c)$. Therefore, $f(U) \geq f(U^c)$. ∎

Since each solution of P_g is either covered or uncovered, a localization result follows.

<u>Corollary 3.3.2</u>. The <u>union</u> of the set of <u>covered solutions</u> of P_g and the set of <u>neighboring intersection solutions</u> contains an optimal solution to P_g.

Furthermore, since all the intersection solutions are uncovered, from Corollary 3.3.2., we know that each intersection solution is either a neighborhood intersection solution or is dominated by some intersection solution. Thus,

<u>Corollary 3.3.3</u>. The set of neighboring intersection solutions contains a best intersection solution.

Corollary 3.3.3 helps reduce considerablly the effort of finding a best intersection solution. Experiment later in this chapter shows that large percent of instances of P_g has the best intersection solutions as the globally optimal solution. We will give a polynomial-time algorithm for the best intersection solution, in subsection 3.2.2.

3.1.4. A Worst-Case Analysis of the Gap $f(U^*) - h(Z^*)$

The following property and the example afterward give some insight into the quality of the approximation. Let L be the longest grid edge length in N_g. Let W_j be the total weight of N_g associated with new facility j and let W_{NN} be the total interaction weights of P_g.

Property 3.4. $f(U^*) - h(Z^*) \le (\Sigma_j W_j + 2 W_{NN}) L$.

Proof. Let Z' be the solution with each z_j' the closest intersection point of z_j^*. Then,

$$h(Z') - h(Z^*) = \Sigma_{ij} w_{ij} [r(v_i, z_j') - r(v_i, z_j^*)] + \Sigma_{j<k} v_{jk}[r(z_j', z_k') - r(z_j^*, z_k^*)]$$
$$\le \Sigma_{ij} w_{ij} [r(v_i, z_j') - r(v_i, z_j^*)] + \Sigma_{<j,k>\in S} v_{jk}[r(z_j', z_k') - r(z_j^*, z_k^*)],$$

where S is the set of new facility pairs <j, k> such that $r(z_j', z_k') > r(z_j^*, z_k^*)$. Since z_j' is the closest intersection point of z_j^*, $r(z_j', z_j^*) \le L$. Thus, from the triangle inequality, we have

$$r(v_i, z_j') - r(v_i, z_j^*) \le L.$$

$$r(z_j', z_k') - r(z_j^*, z_k^*) \le r(z_j', z_j^*) + r(z_k', z_k^*), \qquad \forall <j, k> \in S,$$

so that

$$r(z_j', z_k') - r(z_j^*, z_k^*) \le 2L, \qquad \forall <j, k> \in S.$$

Thus

$$h(Z') - h(Z^*) \le [\Sigma_j W_j + 2\Sigma_{<j,k>\in S} v_{jk}]L.$$

From the last inequality, the worst-case bound is easily obtained by assuming that every new facility pair in P_g is in S. ∎

The following example gives an instance of MMP which has an approximation gap half of that in the worst-case. This example shows that the worst-case bound given in Property 3.4 is only a constant ratio larger than the tightest bound.

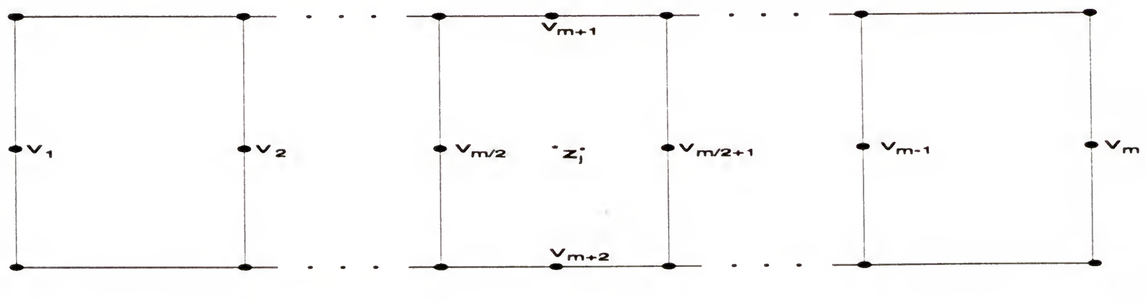


Figure 3.5 A Grid Network of Identical Grid Edge Lengths for a Worst-Case Example

<u>Example 3.3</u>. Let P be an instance defined on the N_g shown in Figure 3.5. The network has identical grid edge lengths of one. Parameters m and n are even integer numbers. The weights of P are defined as follows:

$$v_{jk} = \varepsilon \ \text{ for } \forall j, k; \quad w_{ij} = w,\ i = 1, \ldots, m, j = 1, \ldots, n;$$

$$w_{m+1,j} = \begin{cases} \alpha & \text{if } j \text{ is even} \\ 0 & \text{o/w;} \end{cases} \qquad w_{m+2,j} = \begin{cases} 0 & \text{o/w} \\ \alpha & \text{if } j \text{ is odd;} \end{cases}$$

and $\quad w_{ij} = 0$, for the rest of the vertices (intersection vertices).

Finally, ε, α, and w are positive real numbers such that α/mw is negligible and, in order to have U^* as described below, $n\varepsilon < 2\alpha$.

There is an optimal Z^* with each z_j^* in the center of the area enclosed by N_g, and an optimal U^* with $u_j^* = v_{m+1}$ if j is even and $u_j^* = v_{m+2}$ if j is odd. Therefore, $d(v_i, u_j^*) - r(v_i, z_j^*) = L/2$, for i = 1, ..., m,

$$d(v_{m+1}, u_j^*) - r(v_{m+1}, z_j^*) = \begin{cases} -L/2, & \text{if } j \text{ is even} \\ 3L/2, & \text{if } j \text{ is odd} \end{cases} \qquad d(v_{m+2}, u_j^*) - r(v_{m+2}, z_j^*) = \begin{cases} -L/2, & \text{if } j \text{ is odd} \\ 3L/2, & \text{if } j \text{ is even} \end{cases}$$

and

$$d(u_j^*, u_k^*) - r(z_j^*, z_k^*) = \begin{cases} 0, & \text{if both } j \text{ and } k \text{ are odd, or both } j \text{ and } k \text{ are even,} \\ 2L & \text{otherwise.} \end{cases}$$

Therefore,

$$\begin{aligned} f(U^*) - h(Z^*) &= nmw(L/2) - \Sigma_{\text{even } j}\alpha L/2 - \Sigma_{\text{odd } j}\alpha L/2 + \Sigma_{\text{odd } j}\alpha(3L/2) + \Sigma_{\text{even } j}\alpha(3L/2) \\ &\quad + \Sigma_{\text{even } j}\, \Sigma_{\text{odd } k}\varepsilon(2L) \\ &= nmw(L/2) + 2n\alpha L + 2\varepsilon(n/2)^2 L \\ &= [mw/2 + 2\alpha + \varepsilon n/2]nL. \end{aligned}$$

The worst-case estimation of $f(U^*) - h(Z^*)$ is $[mw + \alpha + \varepsilon n]nL$. The ratio between the real gap and the worst-case estimation is $1/2 + \alpha/[mw + \alpha + \varepsilon n]$. Since α/mw is negligible, the ratio is approximately 1/2.

<u>3.1.5. Summary</u>

In this section, we introduced the following concepts which all related to the structure of N_g in the vicinity of Z^*. They are, the neighborhood of Z^*, the neighboring and nonneighbor solutions, and the covered and the uncovered solutions. We demonstrated through Corollary 3.3.1 that most of the nonneighbor solutions of P_g are dominated by their closest neighboring solutions, so that the set of neighboring solutions of P_g contains near-optimal solutions of P_g. We established analytically that the set of uncovered solutions is dominated. This result leads to a localization result for the best intersection solutions. We also provided a worst-case bound on $f(U^*) - h(Z^*)$.

3.2. Heuristics for Solving P_g

The grid network multimedian problem P_g is NP-hard (Tamir, 1993). From the last section we see that by solving P_r it is possible to identify a solution subset containing a near-optimal solution. But, it is not generally true that such a subset contains an optimal solution. In this section, we discuss some heuristics for searching over this subset, and in the last section of this chapter we discuss the experimental results with these heuristics.

3.2.1. Finding an Approximate Solution to P_g Based on Optimal Solutions to P_r

Heuristic 1. Take U^{I*}, the best intersection solution, as a near-optimal solution.

Later in this section, we will give a simple algorithm to find a U^{I*}. Now, we discuss the insights for Heuristic 1:

a. Solution U^{I*} is the best among all the intersection solutions. With a relatively refined grid network, the intersection solutions should reflect the general trend of the contours of f.
b. Let $U^{I'}$ be a closest intersection solution to Z^*. With a relatively refined grid network, $h(U^{I'}) - h(Z^*)$ should be small. With $h(Z^*) \leq f(U^*) \leq f(U^{I*}) = h(U^{I*}) \leq h(U^{I'})$, the difference $f(U^{I*}) - h(Z^*)$ is smaller.
c. Consider some k variables, say $u_{(1)}, \ldots, u_{(k)}$. Let $g(u_{(1)}, \ldots, u_{(k)}) = f(u_1, \ldots, u_{(1)}, \ldots, u_{(k)}, \ldots, u_n)$ be a function of $u_{(1)}, \ldots, u_{(k)}$ as they vary over a grid edge while other new facilities are fixed. If, in this grid edge, the non-interactive weights (w_{ij}'s) associated with new facilities (1), ..., (k) are insignificant, then g tends to be "concave like". That is, either g is concave or has some "shallow" local minima, so that one of the end points (intersection points) is the minimum or

near minimum of g over the grid edge. This indicates that an intersection solution is usually better than the solutions in its neighbourhood.

d. We can show that if N_g has identical edge (note, not grid edges) lengths and if every new facility j has identical weights w_{ij} (i.e.. $w_{ij} = w_{hj}$ for any h and i), then every best intersection solution is a globally optimal solution. Although this choice of data is uncommon, problems with similar choices of data are common.

The rest of the heuristics below are designed to deal with cases when a best intersection solution is not a near-optimal solution. Let N_j be the neighbourhood of z_j^* (see Definition 3.7) and let V_j be the set of vertices in N_j. Usually, Heuristic 1 fails when there are some non-intersection vertices with large weights. Thus, we design the following heuristics which search the interior vertices in $V_1 \times \ldots \times V_n$.

The idea of heuristic 2 is the following. For a given Z^*, construct U^0 with u_j^0 the vertex in V_j that is closest to z_j^*. For each iteration t, we choose a new facility j and search for an adjacent vertex, say, $v \in V_j$, of u_j^t such that f will decrease if we move new facility j from u_j^t to v while the locations of other new facilities are not changed. The process terminates if no such adjacent vertex can be found for every new facility.

Heuristic 2. (Neighbourhood One Dimensional Search)

Step 0. For the given Z^*, construct $U^0 \in V_1 \times \ldots \times V_n$ with u_j^0 the closest vertex in V_j to z_j^*;

Let $L = \{j \mid |V_j| > 1\}$ and let $L' = L$;

$t = 1$;

Step 1. If $L' = \varnothing$ then terminate the search;

Otherwise, choose a $j \in L'$ and construct $NV_j^t = \{v_i \mid v_i$ is adjacent to u_j^t and $v_i \in V_j\}$;

Step 2. If $NV_j^t = \varnothing$ then let $L' = L' - \{j\}$ and go to Step 1;

Otherwise, choose a vertex $v \in NV_j^t$ and let $u_j' = v$, $u_k' = u_k^t$ for every $k \neq j$;

Step 3. If $f(U') < f(U^t)$ then go to Step 4;

Otherwise, let $NV_j^t = NV_j^t - \{v\}$ and go to Step 2;

Step 4. Let $NV_j^{t+1} = \{v_i \mid v_i \text{ is adjacent to } v \text{ and } v_i \in V_j\} - \{u_j^t\}$ and $NV_k^{t+1} = NV_k^t$, for all $k \neq j$;

Let $U^{t+1} = U'$ and Let $L' = L$;

Let $t = t+1$ and Go to Step 2;

A more thorough search over $V_1 \times \ldots \times V_n$ is to update an intermediate solution U^t by replacing some u_j^t with an optimal vertex solution of problem P_j^t: Minimize $\{ g(u_j) = f(\ldots, u_{j-1}^t, u_j, u_{j+1}^t, \ldots) \mid u_j \in V_j\}$. Such a process repeats itself for different j in each iteration until $U^t = U^{t+1}$ for some t. Each P_j^t is solved by enumeration of V_j.

<u>Heuristic 3</u>. (Local Optimal One Dimensional Search)

Step 0. For the given Z^*, construct $U^0 \in V_1 \times \ldots \times V_n$ with u_j^0 the closest vertex in V_j to z_j^*;

Let $L = \{j \mid |V_j| > 1\}$ and let $L' = L$; $t = 1$;

Step 1. If $L' = \varnothing$ then terminate the search;

Otherwise, choose a $j \in L'$;

Step 2. Let U' be the solution with $u_k' = u_k^t$, for all $k \neq j$, and u_j' be an optimal solution of

P_j^t: Minimize $\{ g(u_j) = f(\ldots, u_{j-1}^t, u_j, u_{j+1}^t, \ldots) \mid u_j \in V_j\}$;

Step 3. If $f(U') < f(U^t)$ then go to Step 4;

Otherwise, let $L' = L' - \{j\}$ and go to Step 1;

Step 4. Let $U^{t+1} = U'$ and Let $L' = L$;

Let $t = t+1$ and go to Step 1;

These two heuristics change one location at a time. It is well-known that optimal locations of MMP tend to coincide. Therefore, it is often futile to change only one location of a set of identical locations. Thus, we design a heuristic which treats each cluster of new facilities as a "super" new facility. The heuristic uses an output solution of one of the above search heuristics as input. Let U^0 be such a solution and let $a_1, \ldots, a_p$ be the distinct locations in U^0.

<u>Heuristic 4</u>. (Super New Facility One Dimensional Search)

Step 0. Construct a new MMP, SP, in which each new facility represents all the new facilities with the same locations in U^0. In SP, $w_{i\alpha}' = \Sigma\{w_{ij} \mid u_j^0 = a_\alpha\}$, $i = 1, \ldots, m$, $\alpha = 1, \ldots, p$, and

$$v_{\alpha\beta}' = \Sigma\{v_{jk} \mid u_j^0 = a_\alpha \text{ and } u_k^0 = a_\beta\},\ 1 \le \alpha < \beta \le p.$$

Step 1. Construct super-neighborhood $V' = V_1' \times \ldots \times V_p' \ni V_\alpha' = \cup\{V_j \mid u_j^0 = a_\alpha\}$, $\alpha = 1, .., p$.

Step 2. With initial solution $U^{0'} = (a_1, \ldots, a_p)$, perform one dimensional search with either Heuristic 2 or Heuristic 3.

3.2.2. Solving P_g^I – the Intersection Restricted P_g

Recall that for a given Z^*, $NIP = NIP_1 \times \ldots NIP_n$ denotes the set of neighboring intersection solutions, where each $NIP_j = \{(lz_j^*, bz_j^*), (lz_j^*, tz_j^*), (rz_j^*, bz_j^*), (rz_j^*, tz_j^*)\}$. From Corollary 3.3.3, there exists a best intersection solution in NIP. Thus, an equivalent formulation of P_g^I is Minimize $\{f(U) \mid U \in NIP\}$. Since every solution in P_g^I is an intersection solution, P_g^I is equivalent to P_r^I: Minimize $\{h(Z) \mid Z \in NIP\}$ which can be decomposed into

P_{rx}^I: Minimize$\{h_x(Z_x) \mid z_{xj} \in \{lz_j^*, rz_j^*\}, j = 1, \ldots, n\}$ and

P_{ry}^I: Minimize$\{h_y(Z_y) \mid z_{yj} \in \{bz_j^*, tz_j^*\}, j = 1, \ldots, n\}$.

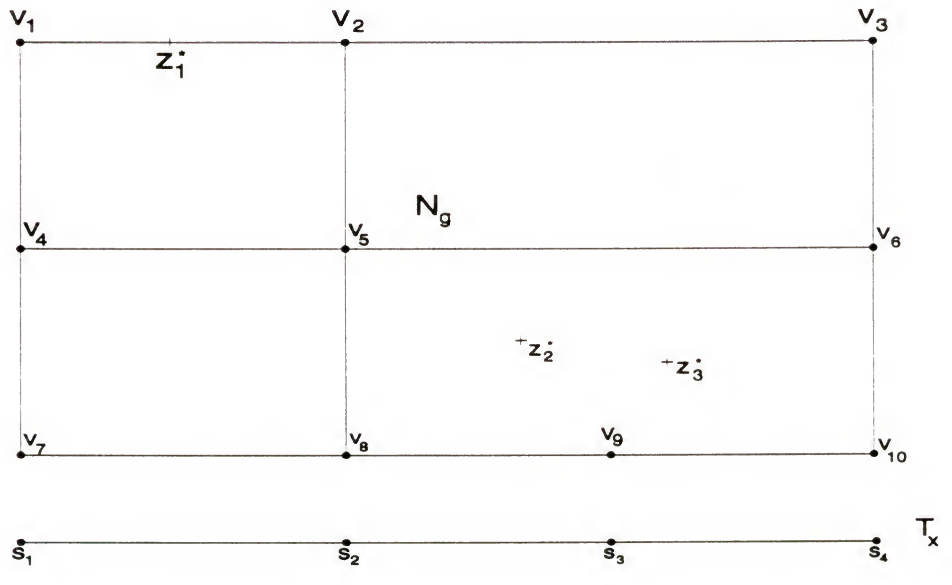


Figure 3.6 The Grid Network in Decomposition Examples

We only need to show how to solve P_{rx}^I, since P_{ry}^I is the same as P_{rx}^I except for notation. First, we can eliminate every new facility j in P_{rx}^I with $lz_j^* = rz_j^*$ from further consideration, by fixing new facility j at the position and modifying the weights accordingly. Thus, we only consider P_{rx}^I with $lz_j^* < rz_j^*$ for every j. Observe that, in P_{xr}^I, each new facility is localized to a block subnetwork the grid interval $[lz_j^*, rz_j^*]$, of the path network. Let $[lz_{(t)}^*, rz_{(t)}^*]$, t = 1, ..., p, be the distinct localized blocks. From the results in Chapter 2, P_{rx}^I can be decomposed into p independent MMP subproblems each of which corresponds to a block.

Example 3.4. Consider an instance of MMP of 3 new facilities on the grid network shown in Figure 3.6. Suppose an optimal solution Z^* has z_1^* an interior point of grid edge (v_1, v_2) and has z_2^*, z_3^* interior points of the rectangle with corner points v_5, v_6, v_8, and v_{10}. From Corollary 3.3.3, the neighboring intersection solution set $\{v_1, v_2\}\times\{v_5, v_6, v_8, v_{10}\} \times \{v_5, v_6, v_8, v_{10}\}$ contains a best intersection solution. Therefore, the solution sets of P_{xr}^I is $\{s_1, s_2\} \times \{s_2, s_3, s_4\} \times \{s_2, s_3, s_4\}$, where s_i's are the four vertices of path network T_x. Problems P_{xr}^I can be formulated as a MMP problem on T_x in the following:

$$P_{xr}^I\text{: Minimize } \Sigma\{\omega_{i1}d(s_i, z_{x1}) \mid i = 1, ..., 4\} + \Sigma\{\omega_{ij}d(s_i, z_{xj}) \mid i = 1, ..., 4, j = 2, 3\} + v_{12}d(z_{x1}, z_{x2}) + v_{13}d(z_{x1}, z_{x3}) + v_{23}d(z_{x2}, z_{x3})$$

$$z_{x1} \in \{s_1, s_2\}$$
$$z_{x2}, z_{x3} \in \{s_2, s_4\}$$

We see that the solution set of P_{xr}^I implies that z_{x1} is localized to block $[s_1, s_2]$ and z_{x2} and z_{x3} are localized to block $[s_2, s_4]$. First,

$$d(s_i, z_{x1}) = d(s_i, s_2) + d(s_2, z_{x1}), \quad \text{for } i = 3, 4, \tag{3.5}$$

and

$$d(s_1, z_{xj}) = d(s_1, s_2) + d(s_2, z_{xj}), \tag{3.6}$$

$$\text{for } j = 2, 3$$

$$d(z_{x1}, z_{xj}) = d(z_{x1}, s_2) + d(s_2, s_2) + d(s_2, z_{xj}). \tag{3.7}$$

Replace, in P_{xr}^I, each distance on the left hand side with its right hand side and rearrange the distance terms,

$$P_{xr}^I\text{: Minimize } \{h_{(1)}(z_{x1}) \mid z_{x1} \in \{s_1, s_2\}\} + \{h_{(2)}(z_{x2}, z_{x3}) \mid z_{xj} \in \{s_2, s_3\}, j = 2, 3\} + C$$

where $h_{(1)}(z_{x1}) = \omega_{11}'d(s_1, z_{x1}) + \omega_{21}'d(s_2, z_{x1})$,

$h_{(2)}(z_{x2}, z_{x3}) = \Sigma\{\omega_{2j}'d(s_2, z_{xj}) + \omega_{3j}'d(s_3, z_{xj}) + \omega_{4j}'d(s_4, z_{xj}) \mid j = 2, 3\} + v_{23}d(z_{x2}, z_{x3})$,

$\omega_{21}' = \omega_{21}+\omega_{31}+\omega_{41}+v_{12}+v_{13}$, $\omega_{2j}' = \omega_{1j}+\omega_{2j}+v_{1j}$, j = 2, 3, $\omega_{ij}' = \omega_{ij}$ for the other i, j,

and C is a constant involving those constant distances in (3.5) to (3.7). Clearly, P_{xr}^I can be decomposed into two independent problems, P': Minimize $\{h_{(1)}(z_{x1}) | z_{x1} \in \{s_1, s_2\}\}$ and P": $\{h_{(2)}(z_{x2}, z_{x3}) | z_{xj} \in \{s_2, s_4\}, j = 2, 3\}$.

We now address solving these subproblems. Let P^I be such a subproblem on the localizing block [lz, rz] which contains vertices $v_1, ..., v_m$, $v_1 = lz$, $v_m = rz$, $v_i < v_{i+1}$. P^I can be expressed as

P^I: Minimize $\{h_x(Z_x)' | z_{xj} \in \{v_1, v_m\}, j = 1, ..., n\}$,

where h_x' is obtained from h_x by weight adjustment as discussed in Chapter 2 as well as in the above example. If [lz, rz] is an edge network (i.e m = 2), then the problem becomes finding an optimal 2-partition $\{J_l, J_r\}$ of new facility indices such that there is an optimal solution in which new facilities in J_l (J_r) are located on v_1 (v_m). We can use Kolen's search algorithm (1982) to determine such an optimal partition. If [lz, rz] contains more than two vertices, we construct an equivalent multimedian problem P on an edge network with only vertices v_s and v_t and with edge length $v_m - v_1$, and solve P with Kolen's search algorithm.

Example 3.5. Consider the subproblem P" in Example 3.4. Since in any solution to P", new facility location z_{xj} is either s_2 or s_4, we know that

$$d(s_3, z_{xj}) = \frac{d(s_2, s_3)}{d(s_2, s_4)} d(s_4, z_{xj}) + \frac{d(s_4, s_3)}{d(s_2, s_4)} d(s_2, z_{xj}).$$

By replacing $d(s_3, z_{xj})$ in P" with the right hand side above, we have

P": $\Sigma_{j=2,3}\{\alpha_{2j}\, d(s_2, z_{xj}) + \alpha_{4j}\, d(s_4, z_{xj})\}$, where

$$\alpha_{2j} = \omega_{2j}+\omega_{3j} \frac{d(s_3, s_4)}{d(s_2, s_4)} \quad \text{and } \alpha_{4j} = \omega_{4j} + \omega_{3j} \frac{d(s_3, s_2)}{d(s_2, s_4)}.$$

In this way, we transform P" into a MMP problem on an edge (s_2, s_4).

In general, we construct problem P as follows. Let $d(v_s, v_t) = d(v_1, v_m)$. Define the weights $\{\alpha_{sj}\}$ ($\{\alpha_{tj}\}$) on the distance between v_s (v_t) and new facility j as

$$\left.\begin{aligned} \alpha_{sj} &= w_{1j} + \Sigma\{w_{ij} \frac{d(v_i, v_t)}{d(v_s, v_t)} | i = 2, .., m-1\}, \\ \alpha_{tj} &= w_{mj} + \Sigma\{w_{ij} \frac{d(v_i, v_s)}{d(v_s, v_t)} | i = 2, .., m-1\}. \end{aligned}\right. \qquad j = 1, ..., n.$$

Since each solution of P^I has each new facility located on either v_1 or v_m, and from the way the weights of P are constructed, for each solution of P^I there is a solution to P which has the same objective value. Thus, P^I and P are equivalent.

Kolen's algorithm takes $O(n^3(m-1))$ steps to solve a MMP with n new facilities and m vertices. Suppose that $P_{rx}{}^I$ is decomposed into k subproblems of new facilities $n_1, ..., n_k$ respectively. Then, it needs $O(n_i{}^3)$ steps to solve subproblem i since subproblem i is a MMP on a network of two vertices. Hence, after $P_{rx}{}^I$ is decomposed, it needs at most $O(n^4)$ to solve $P_{rx}{}^I$. Decomposing $P_{rx}{}^I$ needs O(nm) steps. In actual implementation, we do not need to decompose $P_{xr}{}^I$ explicitly.

3.3. Solving P_g with Branch and Bound

Since P_g is NP-hard, we use a branch and bound approach when the heuristic results are unsatisfactory. On the other hand, we need experience with the relations between P_g and P_r for large problems. Currently, the branch and bound approach is the only feasible exact method for solving a large P_g. The computational experience gained here may also be of help for the general cyclic multimedian problem.

This section has three subsections. Subsection 1 defines the subproblems in the branch and bound process and the initial solution set. Subsection 2 discusses the branching strategy. Subsection 3 shows that a lower bounding problem can be solved efficiently with the solution information obtained from the parent subproblem.

3.3.1 Subproblems and the Initial Solution Set

Let $P_g{}^t$ denote the *t*th node in the branching tree, where $P_g{}^0 = P_g$. Subproblem $P_g{}^t$ is a MMP with solutions restricted to $V^t \equiv V_1{}^t \times ... \times V_n{}^t$. Each $V_j{}^t$ is the set of vertices in subnetwork $N_g \cap R_j{}^t$, where $R_j{}^t = \{z \in N_r \mid lb_j{}^t \leq z_x \leq rb_j{}^t, bb_j{}^t \leq z_y \leq tb_j{}^t\}$ is either a rectangle, a line segment, or a point in N_r.

From Section 3.1, we know that for a given optimal solution Z^* to P_r there exists a near-optimal solution of P_g in the vicinity of Z^*. Thus, the initial solution set is a neighborhood R^0 of

Z^*. To use a simple procedure to construct an initial solution set as small as possible, we choose to solve a series of rectilinear multimedian problems. Let U^0 be a feasible solution of P_g obtained by some heuristics in Section 3.2. Let $P_r(u_{xj} \le a)$ be the resulting problem obtained by adding an inequality $u_{xj} \le a$ to P_r for some constant a. Let V_x (V_y) denote the set of distinct x-coordinates (y-coordinates) of the vertices of N_g. Then,

$$R_j^0 = \{z \mid lb_j^0 \le z_{xj} \le rb_j^0,\ bb_j^0 \le z_{yj} \le tb_j^0\}, \text{ where}$$

$$lb_j^0 = \text{minimum}\{a \in V_x \mid a \le z_{xj}^* \text{ and } \text{obj}(P_r(u_{xj} \le a)) < f(U^0)\},$$

$$rb_j^0 = \text{maximum}\{b \in V_x \mid b \ge z_{xj}^* \text{ and } \text{obj}(P_r(u_{xj} \ge b)) < f(U^0)\},$$

$$bb_j^0 = \text{minimum}\{a \in V_y \mid a \le z_{yj}^* \text{ and } \text{obj}(P_r(u_{yj} \le a)) < f(U^0)\},$$

$$tb_j^0 = \text{maximum}\{b \in V_y \mid b \ge z_{yj}^* \text{ and } \text{obj}(P_r(u_{yj} \ge b)) < f(U^0)\},$$

$$V_j^0 = \{v_i \mid v_i \in R_j^0\}.$$

Set $V^0 \equiv V_1^0 \times \ldots \times V_n^0$ contains an optimal solution to P_g, since each R_j^0 contains an optimal location of new facility j. To see the latter, observe that $\text{obj}(P_r(u_{xj} \le a))$ is a non-increasing function of parameter a. Hence, for any solution U of P_g with $u_{xj} < lb_j^0$, since it is feasible to problem $P_r(u_{xj} < lb_j^0)$ and $\text{obj}(P_r(u_{xj} < lb_j^0)) \ge f(U^0)$, we have $h(U) \ge \text{obj}(P_r(u_{xj} < lb_j^0)) \ge f(U^0)$. From $f(U) \ge h(U) \ge f(U^0)$, solution U can be eliminated from further consideration. Similarly, we can eliminate all those solutions with $u_{xj} > rb_j^0$, or $u_{yj} < bb_j^0$, or $u_{yj} > tb_j^0$.

3.3.2 Branching Strategy

Let P_g^t be the branching subproblem with branching variable u_j. Let c be the number of branching nodes generated so far. The branching strategy is to find a partition, say, $\{V_j^{c+1}, \ldots, V_j^{c+\gamma}\}$ of V_j^t and generate nodes $P_g^{c+1}, \ldots, P_g^{c+\gamma}$ with u_j restricted to $V_j^{c+1}, \ldots, V_j^{c+\gamma}$ respectively.

We construct a partition of V_j^t by partitioning subnetwork $R_j^t \cap N_g$ into some components and let each V_j^{c+k} be the set of vertices in one such component. If R_j^t is a nondegenerate rectangle, we choose to partition it into the grid line segments inside R_j^t, so that relatively few but more different subproblems will be generated. If R_j^t is already a grid line segment, we either partition it into two segments or choose certain subset of vertices inside R_j^t as the partition.

<u>The Branching Procedure</u>:

If R_j^t is a rectangle then

Begin

Partition R_j^t into $\{R_j^{c+1}, ..., R_j^{c+\gamma}\}$ with R_j^{c+k} a grid line segments inside R_j^t; Let

$$V_j^{c+k} = \begin{cases} \{v_i \in N_g \mid v_i \in R_j^{c+k}\}, & \text{if } R_j^{c+k} \text{ is a vertical grid line segment,} \\ \{v_i \in N_g \mid v_i \in R_j^{c+k} \text{ and } v_i \text{ is not an intersection point}\} & \text{if } R_j^{c+k} \text{ is a horizontal grid line segment;} \end{cases}$$

Generate $P_g^{c+1}, ..., P_g^{c+\gamma}$ with u_j restricted to $V_j^{c+1}, ..., V_j^{c+\gamma}$ respectively;

End;

If R_j^t is a vertical grid-line segment [a, b] then

Begin

If the number of vertices in [a, b] is less than a parameter *nv*, then

generate subproblems each of which has u_j restricted to a vertex in [a, b].

Otherwise

Begin

If the length of [a, b] is larger than a parameter L_1 then generate two subproblems with u_j restricted to [a, mp] and [mp, b] respectively, where mp is the vertex in [a, b] closest to the midpoint of [a, b];

Otherwise, generate a subproblem with u_j fixed on a.

End;

End;

If R_j^t is a horizontal grid-line segment [a, b] then

Perform partitioning similar to the case when R_j^t is a vertical grid-line segment.

Figure 3.7 illustrats partitions under different cases. Region R_1^t is partitioned into the grid line segments inside it and region R_2^t is partitioned into two segments. Since region R_3^t is long enough but contains few vertices, it is partitioned into three components each of which is a vertex inside R_3^t. Finally, since the length of region R_4^t is short enough, only one end point is selected as the next candidate location. Finally, we see that a subproblem is generated from P_g^t by adding a set of equalities and/or inequalities of (z_{xj}, z_{yj}) to the set of constraints of R_j^t. For later discussion, we describe the branching strategy in terms of the polyhedrons of Z as the following.

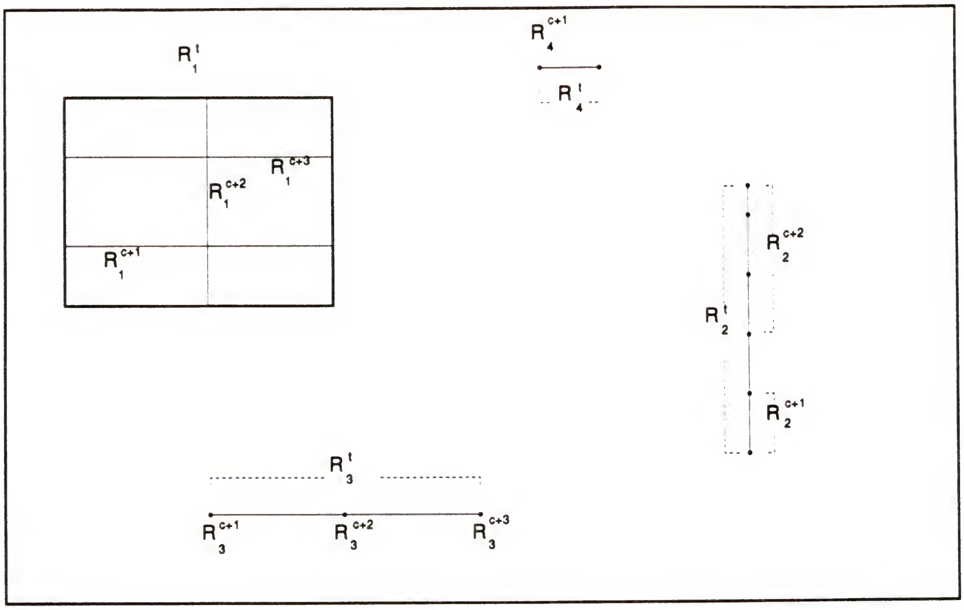


Figure 3.7 Different Partitions

<u>The Branching Procedure</u>:

If, in R_j^t, $lb_j^t < rb_j^t$ and $bb_j^t < tb_j^t$, then

Begin

With $vl_{x1}, ..., vl_{xp}$, $(hl_{y1}, ..., hl_{yq})$ the x-coordinates (y-coordinates) of the vertical grid lines (the horizontal grid lines) such that $lb_j^t \le vl_{xi} \le rb_j^t$ $(bb_j^t \le hl_{yi} \le tb_j^t)$,

Let $R_j^{t+k} = R_j^t \cap \{(z_{xj}, z_{yj}) \mid z_{xj} = vl_{xk}\}$, $k = 1, ..., p$,
$V_j^{c+k} = \{v_i \mid (v_{xi}, v_{yi}) \in R_j^{c+k}\}$, $k = 1, ..., p$;
$R_j^{c+p+k} = R_j^t \cap \{(z_{xj}, z_{yj}) \mid z_{yj} = hl_{yk}\}$, $k = 1, ..., q$;
$V_j^{c+p+k} = \{v_i \mid (v_{xi}, v_{yi}) \in R_j^{c+k}$, v_i is not an intersection vertex$\}$, $k = 1, ..., q$;

Generate subproblems $P_g^{c+1}, ..., P_g^{c+p+q}$ corresponding to $V_j^{c+1}, ..., V_j^{c+p+q}$ respectively;

End;

If $R_j^t = \{(z_{xj}, z_{yj}) \mid a \le z_{xj} \le b, z_{yj} = hl_{yq}$ for some q$\}$ for some $a < b$, then

Begin

If the number of vertices in [a, b] is less than a parameter *nv*, then

Begin

With $V_j^t = \{v_{(1)}, ..., v_{(p)}\}$,

Generate $P_g^{c+1}, ..., P_g^{c+p}$ corresponding to $V_j^{c+1}, ..., V_j^{c+p}$ respectively,

where $V_j^{c+k} = \{v_{(k)}\}$ $(= R_j^{c+k})$;

End

Else

If the length of [a, b] is larger than a parameter L_1 then

Begin

Generate P_g^{c+1} and P_g^{c+2} corresponds to $R_j^{c+1} = \{(z_{xj}, z_{yj}) | a \le z_{xj} \le mp,$ $z_{yj} = hl_{yq}\}$ and $R_j^{c+2} = \{(z_{xj}, z_{yj}) | mp \le z_{xj} \le b, z_{yj} = hl_{yq}\}$ respectively;

End;

Else

Generate P_g^{c+1} with $R_j^{c+1} = \{(z_{xj}, z_{yj}) \mid z_{xj} = a, z_{yj} = hl_{yq}\}$;

End;

If $R_j^t = \{(z_{xj}, z_{yj}) \mid a \le z_{yj} \le b, z_{xj} = vl_{xp}$ for some p$\}$ then

Perform partitioning similar to the case when R_j^t is a vertical grid-line segment.

3.3.3 The Rectilinear Lower Bounding Problem

Recall that for a given subproblem P_g^t, P_r^t denotes the lower bounding problem in which all the underestimates are the rectilinear type. Problem P_r^t can be decomposed into

P_{xr}^t: Minimize $\sum\{w_{ij}|v_{xi} - z_{xj}| \mid (i, j)\} + \sum\{v_{jk}|z_{xj} - z_{xk}| \mid (j, k)\}$
$z_{xj} \in R_{xj}^t$, j not fixed

and

P_{yr}^t: Minimize $\sum\{w_{ij}|v_{yi} - z_{yj}| \mid (i, j)\} + \sum\{v_{jk}|z_{yj} - z_{yk}| \mid (j, k)\}$
$z_{yj} \in R_{yj}^t$, j not fixed.

Since the two problems are identical except for notation, we only need to study P_{xr}^t. With the restrictions R_{xj}^t's, **problem P_{xr}^t is a multimedian problem on a path network T_x with each new facility j restricted to a connected subnetwork (an interval) R_{xj}^t** = $[lb_j^t, rb_j^t]$. If we denote P_R as a class of tree network multimedian problems in which some new facilities are restricted to subtrees, then P_{xr}^t is a special case of P_R where the tree network is a path. We will first give an algorithm, called the restricted search algorithm, to solve P_R, and then show that we do not need to solve P_{xr}^t starting from scratch. In the restricted search algorithm, we assume without loss of generality that each subtree to which a new facility is restricted has all its tip nodes coinciding with the vertices of the tree network.

The algorithm is a modified version of Kolen's direct search algorithm for tree multimedian problems. Let RP denote an instance of the restricted multimedian problem defined on a tree network T, with each new facility j restricted to a subtree T_j. Without loss of generality, we assume that all the tip points of T_j, j = 1, ..., m, are vertices of T. First of all, we modify the optimality condition given by Kolen (1982) for the restricted tree network multimedian problem.

Theorem 3.1. A vertex solution X to RP is optimal if and only if there is no subset of new facilities which can be moved to an adjacent vertex such that, a. no restrictions are violated, and b. the objective value is decreased.

Proof. The necessity is obvious. To prove the sufficient condition, suppose such a solution X is not optimal. Since all the tips of every restricting subtree are vertices, there exists a vertex-optimal solution X^* to RP. Since both x_j and x_j^* belong to subtree T_j, the unique path connecting x_j with x_j^* is in T_j. Hence, set $C = \{Z \mid Z \equiv \lambda X + (1-\lambda)X^*, 0 \le \lambda \le 1\}$ is a feasible solution set to RP (the convex combination is defined in Dearing et al. (1976)). Since $X \ne X^*$, C contains more solutions than X and X^*. The rest of the proof is the same as that in Kolen (1982). That is, since $C - \{X, X^*\}$ is non-empty and $f(X^*) < f(X)$, there exists an edge e along which we can move a subset of new facilities from one end point to the other to obtain another feasible solution in C with objective function value smaller than f(X). This result is contradictory to the sufficient condition, which proves the theorem. ∎

With this optimality condition, we now modify the direct search algorithm given by Kolen (1982) for the restricted tree multimedian problem. In each iteration of this modified algorithm, the additional work is to identify two subsets of free new facilities, which must be located respectively in the two subtrees because of the restrictions. Since the restrictions are on the distances between existing facilities and new facilities, the identification takes O(mn) time in each iteration.

<u>The Restricted Search Algorithm</u>:

Step 0. Let $k = 0$ and $T^k = T$, and let the set of free new facilities be $FF^k = \{1, ..., n\}$;

Step 1. If T^k is a single vertex, then place all the remaining free new facilities at the vertex and terminate the algorithm; Otherwise, go to Step 2;

Step 2. Select a tip vertex, say v_s, of T^k. Let v_t denote the unique vertex, in T^k, that is adjacent to v_s, and let T_s and T_t be the subtrees containing v_s and v_t respectively, where T_s and T_t are obtained from T by removing edge (v_s, v_t) from T. Construct free new facility subsets Q_s and Q_t which, by restriction, must be located on v_s and in T_t respectively (i.e. Q_s (Q_t) contains the indices of those current free new facilities j such that there exists a vertex $v_i \in T_s$ (T_t) with $d(v_i, v_t) > c_{ij}$ ($d(v_i, v_s) > c_{ij}$), where c_{ij} is the upper bound on $d(v_i, v_j)$;

Step 3. Let $P = FF^k \setminus (Q_s \cup Q_t)$. Let X be the location vector with new facilities in $Q_s \cup P$ (Q_t) located at v_s (v_t) and all the fixed new facilities located at their designated vertices. Determine a subset S of P which, when the new facilities in S are moved to v_t, gives the largest decrease in the objective function. Such a subset S is determined by solving a maximum flow problem on a directed network of at most n+2 nodes (Kolen 1982).

Step 4. If $S \cup Q_t = \varnothing$, then terminate the algorithm since the current solution X is optimal; Otherwise, fix new facilities in $Q_s \cup (P \setminus S)$ at v_s, let $FF^{k+1} = Q_t \cup S$, $T^{k+1} = T^k - \{v_s\}$, and $k = k + 1$. Go to step 1.

The optimality proof for the restricted search is the same as that given by Kolen (1982). Compared with the original direct search algorithm, the only additional work for this modified algorithm is to determine the subsets Q_s and Q_t in each iteration. The complexity of this additional work is O(n). Thus, the modified algorithm also has complexity $O(mn^3)$.

We often do not need to solve P_{xr}^t starting from scratch. Suppose that P_g^s is the parent problem of P_g^t and P_{xr}^s is one of the decomposed subproblem of P_r^s. Recall that P_g^t is derived from P_g^s by letting $R_{xj}^t \subseteq R_{xj}^s$, $R_{yj}^t \subseteq R_{yj}^s$, and at least one of R_{xj}^t and R_{yj}^t is a proper subset. If $R_{xj}^s = R_{xj}^t$, then P_{xr}^t is equivalent to P_{xr}^s. If $R_{xj}^t \subset R_{xj}^s$, then we will show in the following property that we only need to consider a subset of the current unfixed new facilities.

Property 3.4. Let Z_x^s be an optimal solution to P_{xr}^s. Suppose $R_{xj}^t = [lb_j^t, rb_j^t] \subset R_{xj}^s = [lb_j^s, rb_j^s]$.

Case 1. If $z_{xj}^s \in R_{xj}^t$ then Z_x^s is also an optimal solution to P_{xr}^t.

Case 2. If $rb_j^t < z_{xj}^s$ then there exists an optimal solution Z_x^t to P_{xr}^t such that $z_{xj}^t = rb_j^t$ and $z_{xk}^t = z_{xk}^s$ if $z_{xk}^s \notin (rb_j^t, z_{xj}^s]$ for any k.

Case 3. If $z_{xj}^s < lb_j^t$ then there exists an optimal solution Z_x^t to P_{xr}^t such that $z_{xj}^t = lb_j^t$ and $z_{xk}^t = z_{xk}^s$ if $z_{xk}^s \notin [z_{xj}^s, lb_j^t)$ for any k.

Proof. The conclusion for Case 1 is obvious. Now, we prove Case 2. Let $t_{x1}, \ldots, t_{xp}$, $t_{xi} < t_{xi+1}$, be the vertices of path network T_x. Recall that the modified direct search algorithm considers an edge of T_x in each iteration. In iteration, say h, it finds a tip vertex, say v^h, of the current tree network, say T^h, and moves a subset of new facilities to the subtree $T^h - \{v^h\}$. We see that both Z_x^s and Z_x^t can be obtained by applying Kolen's modified search algorithm to P_{xr}^s and P_{xr}^t respectively with the tip vertices chosen in the order of $t_{x1}, t_{x2}, \ldots$. Let P^s and P^t denote these two search processes. For the given rb_j^t, let t_{xq} be the vertex such that $t_{xq} = rb_j^t$. Let $L = \{k \mid z_{xk}^s \leq rb_j^t\}$ and $R = \{k \mid z_{xk}^s > z_{xj}^s\}$. We see that $L \cup R = \{k \mid z_{xk}^s \notin (rb_j^t, z_{xj}^s]\}$. Thus, we need to show that $z_{xk}^t = z_{xk}^s$, for any $k \in L \cup R$.

Since $z_{xj}^s > rb_j^t$, the new restriction $lb_j^t \leq z_{xj} \leq rb_j^t$ in problem P_{xr}^t does not make the execution of process P^t any different from the execution of process P^s until the qth iteration, for which t_{xq} $(= rb_j^t)$ is the tip vertex. Thus, both processes produce the same locational decisions in their respective first q−1 iterations. That is, if a z_{xk}^s is determined in the first q-1 iterations of P^s, then z_{xk}^t is also determined in the first q−1 iterations of P^t and $z_{xk}^t = z_{xk}^s$. Since z_{xk}^s, $k \in L$, is determined in the first q−1 iterations of process P^s, we know that $z_{xk}^t = z_{xk}^s$, for any $k \in L$. Also, in each iteration h, $1 \le h \le q-1$, of process P^t, new facility j is always moved to subtree $T^h - \{v^h\}$ as it has been moved in P^s. At the iteration q of P^t, new facility j is fixed at v^q ($= rb_j^t$), because of new restriction $z_{xj} \le rb_j^t$. Thus, we know that $z_{xj}^t = rb_j^t$.

The proof of Case 2 is complete if we can show that $z_{xk}^t = z_{xk}^s$ for every $k \in R$. This can be done by applying the direct search algorithm to both problems with tip vertices chosen in the order $t_{xp}, t_{xp-1}, \ldots$.

The principle of proving Case 3 is the same as that in proving Case 2. ∎

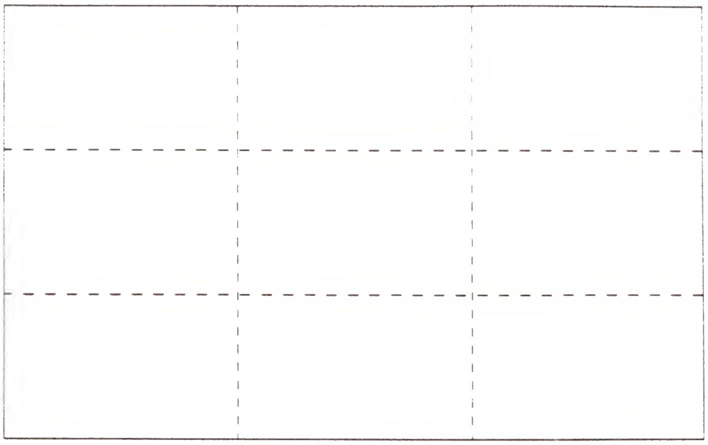
If Case 2 of Property 3.4 is true, then we have a localization result for P_{xr}^t with $z_{xk}^t = z_{xk}^s$ for $k \in L \cup R$, $z_{xj}^t = rb_j^t$, and $z_{xk}^t \in [rb_j^t, z_{xj}^s]$ for any other k. Therefore, we only need to solve a multimedian problem defined on the interval $[rb_j^t, z_{xj}^s]$ with new facilities to be those which are localized to this interval. We can reduce the size of P_{xr}^t in the same way when Case 3 of Property 3.4 is true.

3.4 Computational Experience

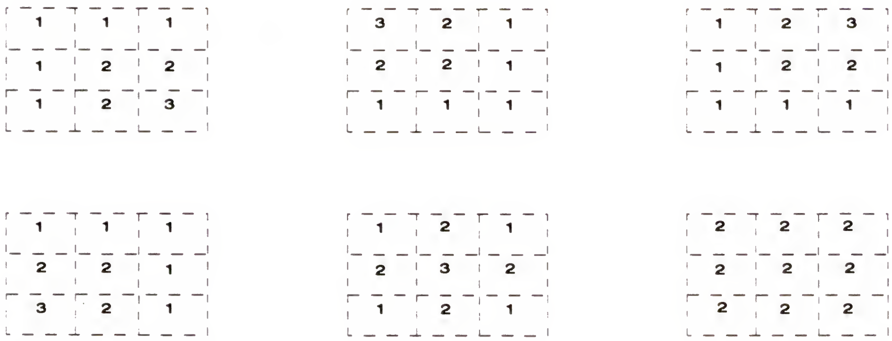
Generally speaking, if N_g has cells of extreme width-height ratios and the weights concentrate on these cells, then the corresponding P_r and their subproblems are not good approximations. At the other extreme, if many of the cells in N_g have width-height ratios close to 1, and N_g is refined (having several grid rows and grid columns), then there should be good lower bounds. We would like to obtain more concrete and more detailed evidence for this intuition on the relationship between the approximation quality and the grid network topologies. We also would like to see the performance of some heuristics, in particular, the intersection optimal solution. In this section, we will discuss the design of experiment, the heuristic considerations, and the computational results.

3.4.1 Experimental Design

We select a spectrum of grid networks of various structures. We generate testing problems by sampling weights $\{w_{ij}\}$ and $\{v_{jk}\}$ from populations of uniform distributions F_{ij} and G_{jk} as follows. Though, it is simpler to have all w_{ij} identically distributed, an instance generated in this way often has many new facilities having similar weight distribution patterns. As a result, many new facilities will coincide. In order to generate more diverse instances, we generate different (weight distribution) patterns for different new facilities. First, we divide the rectangle N_r into six



(a) Zones



(b) Patterns

Figure 3.8 Zones and Patterns

rectangular zones as shown in Figure 3.8a. The weight distribution patterns for two new facilities are different if, in some zones, the weights of one new facility are statistically larger than that of the other. Let $F = \{F_1, F_2, F_3\}$ be set of three uniform distributions with expectations μ_1, μ_2, μ_3, $\mu_1 < \mu_2 < \mu_3$. A distribution pattern is a mapping from zones 1, ..., 6 to F. Let PA be the set of patterns as shown in Figure 3.8b. For each new facility j, we choose a pattern from PA with equal probability and generate the $\{w_{ij}\}$ accordingly. That is, if the distribution in zone l is designated to be F_l, then each weight w_{ij} in zone l is generated from a population having distribution F_l. In actual implementation, F_2 is the uniform distribution in $[a_2, b_2]$ with a_2 and b_2 given. We let F_1 and F_3 be the uniform distributions in intervals $[\max\{0, a_2-K\}, \max\{1, b_2-K\}]$ and $[a_2+K, b_2+K]$ respectively, where $K = \tau(b_2 - a_2)U(0, 1)$ with τ a coefficient given. Intuition and initial experimentation tell us that the algorithm performs relatively well for problems with large interactive weights $\{v_{jk}\}$ (tighter bindings among new facilities makes changing a single location have a greater effect on the objective function. With the algorithm generating subproblems by changing locations of one single new facility at a time, the tighter the bindings among new facilities, the earlier the fathoming occurs). In actual implementation, we let $v_{jk} = a_1 + (b_3 - a_1)(U(0, 1))^3$. The v_{jk}'s generated in this way tend to concentrate more in the neighborhood of a_1. As a final note, this zoning approach generate instances which often have many of the weights concentrated to a subnetwork (a zone). As discussed at the beginning of this section, the performance of the algorithm will be worse on instances like these. Together with the approach in generating interactive weights $\{v_{jk}\}$, we believe that we take a quite conservative approach in generating test problems.

3.4.2 Heuristic Considerations

The algorithm has several important heuristic considerations, in the form of parameters: let UB be the current best upper bound of P_g and $LB(P_g^t)$ be the rectilinear lower bound for P_g^t; the parameters are as:

α The tolerance: Prune P_g^t when the relative gap $(UB - LB(P_g^t)) / LB(P_g^t)$ is less than α;

nv When the number of vertices in R_j^t (the subnetwork to be partitioned to generate subproblems), is less than nv, the algorithm generates subproblems each has the branching new facility fixed to a vertex of R_j^t;

L_1 If R_j^t is a grid-segment with the ratio of its length over the length of N_g in the corresponding dimension less than L_1, then the algorithm only generates two subproblems from P_g^t by fixing new facility j to the two end points of R_j^t;

w_1, w_2 The algorithm chooses, among the current active subproblem, a subproblem, say P_g^t, that has the smallest "chance function value" w_1 / LB(P_g^t) + w_2 S(R^t) as the next branching subproblem. Here S(R^t) is the number of vertex solutions in the solution subset R^t on which P_g^t is defined;

β_1, β_2 After solving P_r^t, the lower bounding problem P_g^t, the algorithm determines whether to use Heuristic 2, the Neighbourhood One Dimensional Search Heuristic, to find a feasible solution to P_g^t, depending on whether P_g^t is different significantly from its parent subproblem. The condition to use Heuristic 2 is

$$(UB - LB(P_g^t)) / LB(P_g^t) > \beta_1 \text{ AND } r(z_j^t, z_j') / L \geq \beta_2$$

where z_j^t and z_j' are the optimal locations of branching new facility j in P_r^t and in the parent problem of P_r^t; and L is either the width or the height of N_g (If the above criteria, is not safisfied, a subproblem is too similar to its parent subproblem. It is unlikely for the Neighborhood One Dimensional Search to find a significantly better solution).

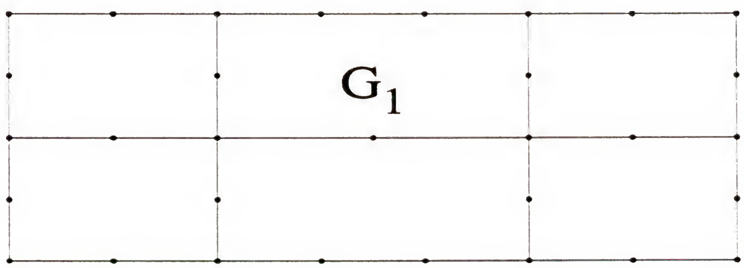


Figure 3.9 Grid Networks Tested: Average-Case

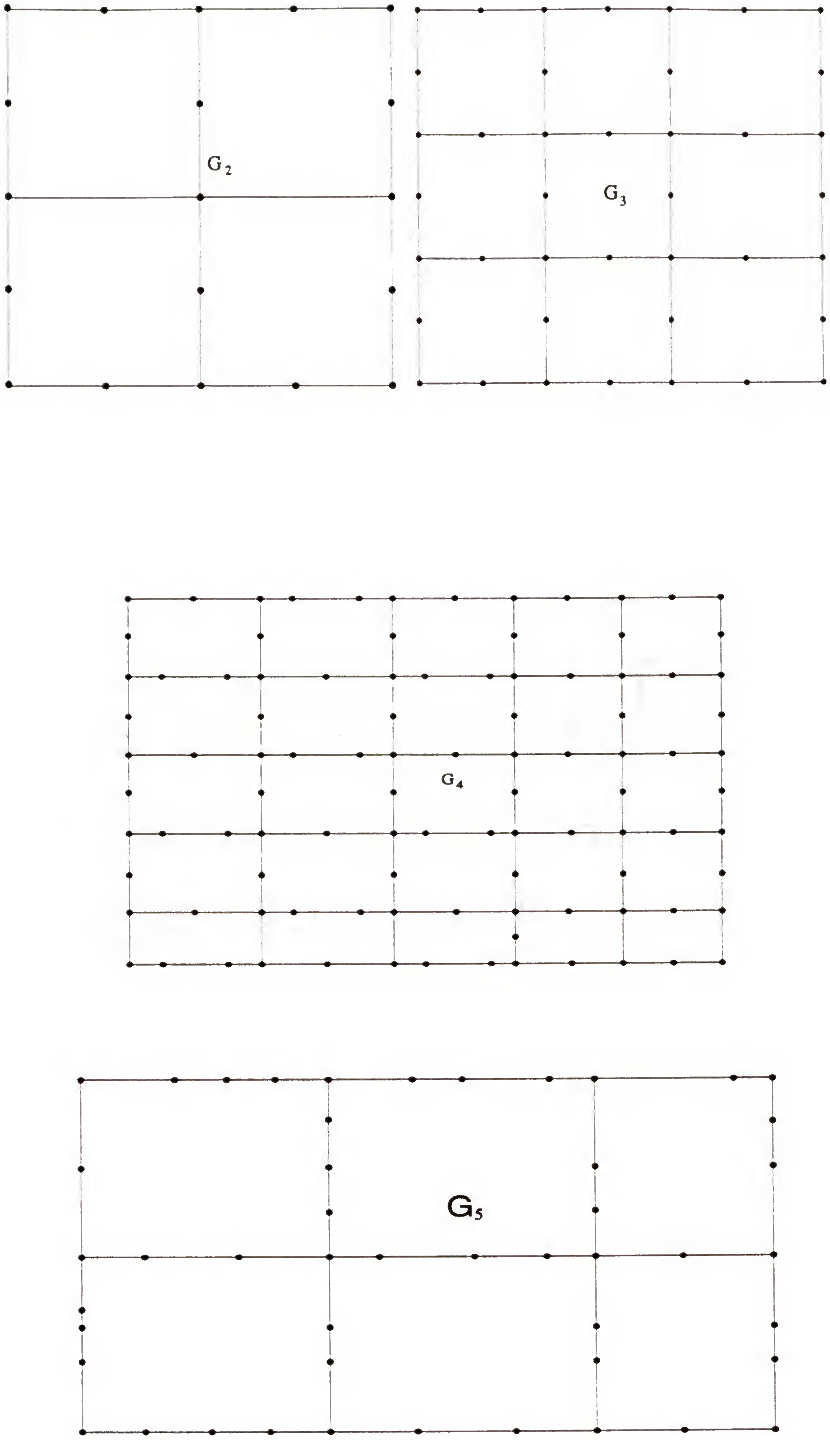


Figure 3.9 Continued

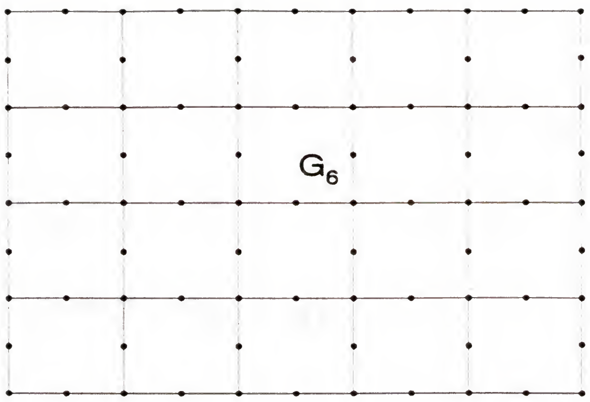


Figure 3.9 Continued

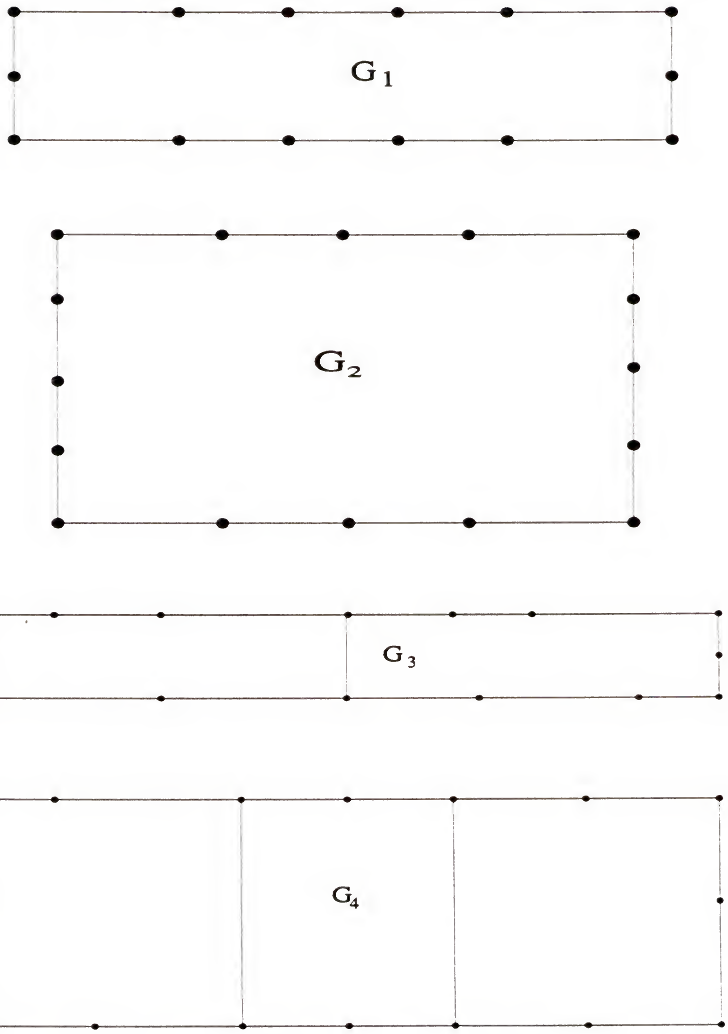


Figure 3.10 Grid Networks Tested: Worst-Case

After initial experiment, we let $nv = 8$, $L_1 = 0.1$, $w_1 = 0.1$, $w_2 = 0.9$, $\beta_1 = 0.3$, and $\beta_2 = 0.3$. We set α differently for different problems, based on the sizes and the topologies of the corresponding problems. These values are shown in columns 11 in Tables 3.1 through 3.10. In most cases, α is zero. The algorithm is sensitive to α and nv, but is not sensitive to w_1 and w_2

3.4.3 Computational Results

We tested ten different grid networks shown in Figure 3.9 and 3.10. The networks in Figure 3.9 have the cell width-height ratio close to one for every cell and have more than one grid-row and one grid-column, while the networks in Figure 3.10 have extreme cell width-height ratios and have only one grid-row and one grid column. Given that weights are generated in the same way, the algorithm will have poorer performance for the problems defined on the latter four networks than for the problems defined on the first six. We call the first six the average-case networks and the latter four the worst-case networks The algorithm is programmed in C and was run on a DEC-5000 RISC computer under Ultrix 4.2. The computer has 80 mega-bytes real memory and has a 20MHz clock which is equivalent to about 24 MIPs. For the average-case networks, we solved20 instances for each different number of new facilities. For the worst-case networks, we solved 10 instances for each different number of new facilities. The results are shown in Tables 3.1 ..., Table 3.10, respectively for the 10 networks. The CPU time is the average one. The IS, BS, Z^*, and U^* denote, respectively, the intersection optimal solution, the best solution found by the algorithm, an optimal solution of P_r, and an optimal solution for P_g. Column 6 in each table summarizes the percentage of instances which have their best solution found before the branch and bound stage. That is, they are found either by Heuristic 1 which solves for an optimal intersection solution or by Heuristic 2, the Neighborhood One Dimensional Search. Column 7 shows the percentage of instances which have their BS = IS. Since it is well-known that the optimal locations for a multimedian problem tend to coincide, we include in Column 8 the percentage of instances in which the new facilities coincide in their BS's. Column 11 give the error tolerances. Column 9 and 10 are the relative errors.

3.4.3.1. Computational Results for the Average-Case

From Tables 3.1 to 3.6, we see that the algorithm is capable of solving most of the instances optimally. The algorithm solves a small percentage of instances sub-optimally, especially the instances defined on network G_3. In G_3 many vertices are in the middle of grid edges - the locations where the rectilinear distance and the grid network distance differ the most. The algorithm performs exceptionally well for instances defined on G_2 which is quite symmetric (so that optimal locations are very likely to be the intersection in the middle of the network). This shows that the algorithm performance is affected significantly by network structures. We are able to solve problems with non-trivial sizes that have never been solved optimally. The largest problem has 30 new facilities on grid network G_4 of 100 vertices. In all the cases tested, the sub-optimal solutions are very close to optimal. The most notable is the quality of the optimal intersection solution. **Nearly all the best solutions are optimal intersection solutions**. For the few exceptions, the relative errors between the optimal intersection solution and the best solution found are too small to bear any significance. Thus, an optimal intersection solution is quite adequate in general. Finally, the computation time increased considerably when we lowered the error tolerance. We believe this is because that the objective function of the multimedian problem has a flat surface, so that there are many solutions having their objective function values close to each other. A higher error tolerance will make the algorithm ignore insignificant objective value differences and start early pruning.

3.4.3.2. Computational Results for the Worst-Case

For the instances on worst-case grid networks, the performance deteriorates considerably. But there are still many instances whose best solutions are found before the branch and bound stage. In comparison to the lower percentage of instances whose optimal intersection solutions are the best solutions, we see that Heuristic 2 is useful here. We believe that the lower bounding problems we propose in the next chapter will be more effective in dealing with problems defined on the worst-case grid networks.

Table 3.1 Average-Case G.1

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	60	220	0.0224	100%	100%	65%	0.0	0.0377	0.0
10	20	399	2529	0.4133	100%	100%	90%	0.0	0.0386	0.0
15	20	4779	39031	10.5382	100%	100%	60%	0.0	0.0447	0.0
20	20	2921	13580	12.7531	100%	100%	85%	0.0	0.0421	0.0
25	20	13659	107951	97.9025	100%	100%	100%	0.0	0.0476	0.0
30	20	2220	9892	23.7535	100%	100%	100%	0.0	0.0463	0.015

Table 3.2 Average-Case G.2

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	38	273	0.0072	100%	95%	95%	0.0002	0.0174	0.0
10	20	46	526	0.0383	100%	100%	100%	0.0	0.0076	0.0
15	20	1	4	0.0004	100%	100%	100%	0.0	0.0004	0.0
20	20	1	1	0.0000	100%	100%	100%	0.0	0.0	0.0
25	20	1	1	0.0000	100%	100%	100%	0.0	0.0	0.0

Table 3.3 Average-Case G.3

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	404	1773	0.0895	100%	100%	50%	0.0	0.0275	0.0
10	20	2702	18524	2.5686	100%	100%	15%	0.0	0.0213	0.0
15	20	10981	75848	31.1386	100%	100%	25%	0.0	0.0274	0.01
20	20	961	14991	6.2404	100%	100%	40%	0.0	0.0268	0.02
25	20	4280	17402	43.2722	100%	100%	50%	0.0	0.0362	0.02
30	20	1730	10989	23.4471	100%	100%	55%	0.0	0.0337	0.02

Table 3.4 Average-Case G.4

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	81	277	0.0258	100%	100%	15%	0	0.0082	0.0
10	20	3180	13294	1.9760	90%	90%	20%	0	0.0082	0.0
15	20	16567	119309	22.2636	85%	85%	10%	0.0001	0.0070	0.0
20	20	5638	71809	10.1230	100%	100%	40%	0	0.0068	0.005
25	20	5426	45728	15.9414	100%	95%	10%	0	0.0083	0.005
30	20	743	8767	4.4124	100%	100%	25%	0	0.0075	0.008

Table 3.5 Average-Case G_5

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	264	1790	0.0508	95%	90%	25%	0.0016	0.0264	0
10	20	5273	73730	3.0519	100%	95%	70%	0	0.0130	0
15	20	1030	13233	1.4667	100%	100%	85%	0	0.0075	0
20	20	10306	114413	10.4210	100%	100%	80%	0	0.0075	0
25	20	365	2842	4.3187	100%	100%	65%	0	0.0090	0.01
30	20	49	973	1.2665	100%	100%	100%	0	0.0055	0.01

Table 3.6 Average-Case G_6

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	70	258	0.0184	95%	95%	35%	0.0001	0.0074	0
10	20	1682	200912	0.7597	95%	95%	45%	0	0.0086	0
15	20	3416	12776	3.4658	100%	100%	50%	0	0.0086	0
20	20	6356	28640	11.6489	100%	100%	75%	0	0.0095	0
25	20	12320	38730	30.6994	100%	100%	80%	0	0.0091	0
30	20	20268	94785	94.4744	100%	100%	75%	0	0.0088	0

Table 3.7 Worst-Case G.1

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
2	10	109	241	0.0226	60%	20%	60%	0.0248	0.2640	0.0
3	10	963	2579	0.1863	60%	10%	10%	0.0431	0.2769	0.0
5	10	83615	343938	21.2850	60%	10%	10%	0.0239	0.2921	0.01
10	10	53516	225854	152.3106	60%	30%	10%	0.0137	0.3152	0.15

Table 3.8 Worst-Case G.2

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
2	10	110	239	0.0223	90%	40%	30%	0.0039	0.1599	0.0
3	10	616	2823	0.1359	80%	50%	20%	0.0048	0.1563	0.0
5	10	12971	50395	8.4912	80%	50%	10%	0.0254	0.1521	0.01
10	10	631	4939	2.8916	100%	90%	40%	0.0005	0.1740	0.15

Table 3.9 Worst-Case G.3

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
2	10	5	16	0.0019	100%	60%	60%	0.0546	0.0792	0
3	10	97	398	0.0551	100%	60%	20%	0.0504	0.0790	0
5	20	68	180	0.0523	90%	65%	20%	0.0162	0.0413	0
10	10	208	1220	0.4468	90%	80%	30%	0.0036	0.0297	0
15	10	5887	33574	18.3176	90%	50%	20%	0.0467	0.0716	0

Table 3.10 Worst-Case G.4

n	No. of Probs. Tested	Ave. #. of Branch Nodes	Max. No. of Branch Nodes	CPU (Sec.s)	%. of Cases where BS < B&B	%. of Cases where BS = IS	% of Sol. with facility Coincide	$\frac{f(IS) - f(BS)}{f(IS)}$	$\frac{f(IS) - h(Z^*)}{f(IS)}$	$\frac{f(BS) - f(U^*)}{f(BS)}$
5	20	335	3022	0.2312	80%	45%	25%	0.0123	0.0430	0.0
10	11	1069	7339	2.5204	82%	64%	19%	0.0185	0.0616	0.0
15	11	9044	50884	45.2855	100%	64%	18%	0.0038	0.0458	0.01
20	10	5278	29067	72.4774	100%	70%	40%	0.0005	0.0434	0.02

CHAPTER 4
LINEARLY CONSTRAINED MULTIFACILITY SUBPROBLEMS AND THEIR PIECEWISE LINEAR AND CONVEX LOWER BOUNDS

In the history of network location analysis, a common approach is to identify a finite dominating solution set and devise algorithms to find an optimal solution in such a set. Also, there is a less developed branch and bound approach for some network location problems which have no known finite dominating set. This approach identifies a particular partition of the solution set, for which the subproblems associated with the elements in the partition are simpler. Since the size of the partition is generally very large, branch and bound techniques are used to find an optimal solution. To our knowledge, this approach has only been applied to problems that involve distance functions each of which is a function of only one location variable (the type-I distance). In this chapter we apply this approach to the following multifacility location problem that involves distance functions of two location variables (type-II distances):

P: $\underset{X \in G^n}{\text{Minimize}}\ f(X) = c(D(X))$.

Here, G is a cyclic network with m vertices, c is a real-valued non-decreasing convex function, and D(X) is a vector $(\ldots, d(v_i, x_j), \ldots, d(x_j, x_k), \ldots)$ of distances. Two special cases of P are the focus of this chapter – the multimedian problem for which $c(.)$ is the sum of weighted distances, and the multicenter problem for which $c(.)$ is the maximum of weighted distances. We give special consideration to these two problems on grid networks.

Instead of giving a complete branch and bound algorithm for a given problem, we concentrate on <u>two important steps</u> – defining solution subsets and hence the subproblems and developing lower bounding techniques (For later reference, we call these two steps a <u>branch and bound scheme</u>). First, we identify a partition Ω of G^n such that over each element of Ω all the distance functions in D(X) are linear. Since Ω is very large, we define solution subsets, called

L-sets, each of which is the union of some elements in Ω. An L-set is closely related to the Cartesian product of some n subnetworks of G, and can be represented as a simple polytope in E^n. Then, we introduce lower bounding techniques for the subproblems defined on L-sets. The lower bounding problems are based on some piecewise linear and convex underestimates of the distance functions in D(X), as well as some piecewise linear and convex underestimates of some components of *c*(D(X)) when *c*( ) is partially separable. The existence and the approximation quality of these underestimates depend on the L-sets. The lower bounding problems are linearly constrained convex programs in general, and are linear programs when *c*( ) is the sum or the maximum of distances.

Our study differs from past work in the choice of location problems, the decomposition strategies, and the subproblems. The notion of using piecewise linear and convex underestimates is new for multifacility network location problems. We believe that our approach is also useful for those multimedian type location problems, since their having dominating finite sets of solutions (the n-fold Cartesian product of the vertex set) has not been known to be utilized in developing any practical optimal algorithms. The existence of dominating solution set certainly should be taken into consideration in designing a detailed branch and bound algorithm for the multimedian type problems.

Throughout this chapter, we will use PLC to refer to piecewise linear and convex. We also provide a glossary of notation in Appendix B.

Now, we give our motivation. Two major difficulties in continuous cyclic network location problems are the non-convexity of the distance functions and their lack of unique analytical form over domains larger than Cartesian products of edge(s). The first difficulty inhibits finding efficient optimal graph-theoretic algorithms. The second difficulty inhibits formulating the problem and its subproblems as mathematical programming problems in E^n. The material presented in this chapter is the first step to tackle these two difficulties. For problems defined on general cyclic networks, the B&B scheme proposed here is useful when the number of new facilities is small. It is particularly useful for those problems, such as the multicenter problem, which have no known finite dominating set, and therefore no known optimal algorithms. With some specialization, the approach given in this chapter will lead to practical B&B algorithms for problems defined on grid networks. Efficient algorithms are expected for the grid-network multimedian and the multicenter problems, which are important location problems having many potential applications in manufacturing, urban planning, and transportation.

The insight for why this scheme should be relatively efficient for the grid-network multimedian and multicenter problems is the following. First of all, from Chapter 3 we see that the rectilinear distances are poor underestimates only when the grid network is sparse (i.e. the grid network has few grid rows and columns). In this case, the grid network can be partitioned into a few components, each of which corresponds to a segment in some grid line. A B&B algorithm thus only needs to consider relatively few subproblems. More importantly, a subproblem with some location variable restrictions (such as some variables restricted to some grid lines) has PLC underestimates of some components of the objective function. These underestimates approach quickly towards their originals in the process of decomposition. The lower bounding problems are linear programs and their sizes can be kept from growing too large by sacrificing certain degrees of quality. On the other hand, when the grid network is "dense", we can solve the problem in two stages. In the first stage, we use another B&B algorithm, which only uses the much easier rectilinear lower bounding problems as in Chapter 3, to find a series of initial solution sets. In each such initial solution set, some "important location variables" are restricted to a few segments of some grid lines. In the second stage, we use the B&B scheme discussed in this chapter to find the best solution inside each of the initial solution sets.

For an overview of this chapter, in Section 1, we introduce some notation and give several examples to highlight principal ideas. In Section 2, we review past results. In Section 3, we define the partition Ω for a cyclic network. In Section 4, we consider multifacility location problems defined on general cyclic networks. In Section 5, we discuss special treatments for problems defined on grid networks. In both Sections 4 and 5, the solution subset S is formally defined, and some technical problems of representing and operating on such a subset are discussed. Procedures for constructing PLC underestimates under various conditions are given. Some techniques for reducing the sizes of lower bounding problems are also discussed. Section 6 summarizes the results.

4.1 Notation and Examples

As a convention, an edge e with vertices u and w is always expressed as (u, w) if u has a smaller index than w. Let [e] and [u, w] both denote the set of points in an edge e = (u, w) including u and w. An arbitrary point x in [u, w] is represented by $t(x)$ -- the length from x to u along edge (u, w). With $t(\ )$, [u, w] is an ordered set. Let CL denote a segment of some edge e, so that [CL] is a subset of [e]. To simplify, we use CL itself to refer to both the segment and the set of points in the segment. With t, there is a one-to-one mapping between a point-vector subset $S \equiv \{(x_1, \ldots, x_p) \mid x_j \in CL_{[j]}, j = 1, \ldots, p\}$ in G^p to a vector subset $t(S) \equiv \{(t(x_1), \ldots, t(x_p)) \mid (x_1, \ldots, x_p) \in S\}$ in E^p. Thus, we can apply the concepts of Euclidian space to S. For example, a vector $(x_1, \ldots, x_p)$ is an extreme point of S if $(t(x_1), \ldots, t(x_p))$ is an extreme point of $t(S)$. A hyperplane in S involving variables $x_1, \ldots, x_p$ is in fact a hyperplane in E^n involving variables $t(x_1), \ldots, t(x_p)$.

With t, we define distance functions as real-valued functions on subsets of E^n. With point-wise location variables x and y restricted to two edges, say (u(x), w(x)) and (u(y), w(y)), the distance function d(x, y) on $[u(x), w(x)] \times [u(y), w(y)]$ corresponds to a real-valued function of $t(x)$ and $t(y)$ on $t([u(x), w(x)] \times [u(y), w(y)])$. We will study type-II distance function d(x, y) in more detail in Section 4.3. We now consider the special case of d(x, y) -- the type-I distance function, when either x or y is fixed at a vertex, say v. Assume that y is fixed at a vertex v and x is in [u(x), w(x)]; It is well-known that

$$d(v, x) = \min\{d(v, u(x)) + t(x), d(v, w(x)) + t(w(x)) - t(x)\} \qquad (4.1).$$

now, d(v, x) is the minimum of two linear functions of $t(x)$ so that it is either linear or is of a "roof-top" shape and so is concave. The <u>antipodal point</u> v^a of v (on edge (u(x), w(x))) is the point where d(v, x) reaches its maximum in [u(x), w(x)]. The function d(v, x) is linear in [u(x), v^a] and [v^a, w(x)] respectively. In general, on each edge, there is exactly one antipodal point for every vertex (Hakimi, 1964), so that there are at most m distinct antipodal points on an edge. An edge then consists of at most m−1 segments over each of which all the type-I distance functions are linear. These segments are called linear segments (Hooker, 1986, 1989).

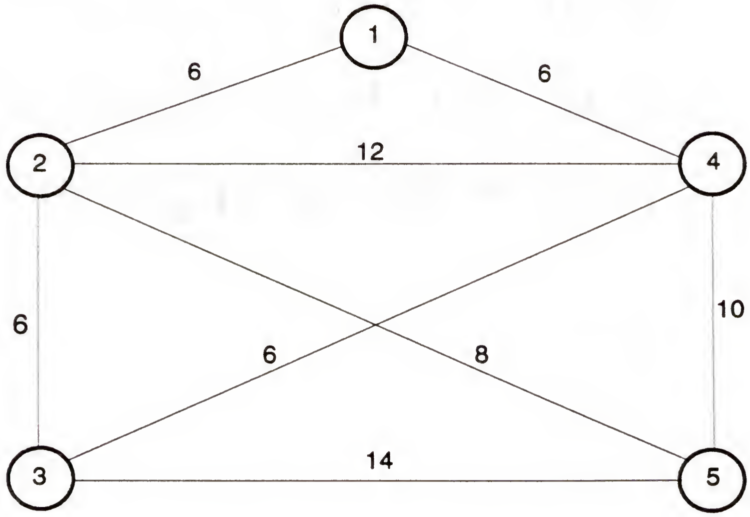


Figure 4.1 A Cyclic Network

Example 4.1. Consider the network in Figure 4.1. Table 4.1 gives the labels and the lengths of edges, and the antipodal points in the interior of each edge.

Table 4.1 Labeling Edges and the Antipodal Points

Edges	Labels	Length	Antipodal Points in the Interior
(v_1, v_2)	e_1	6	
(v_1, v_4)	e_2	6	
(v_2, v_3)	e_3	6	
(v_2, v_4)	e_4	12	v_1^4 ($t(v_1^4) = 6$), v_3^4 ($t(v_3^4) = 6$), v_5^4 ($t(v_5^4) = 7$)
(v_2, v_5)	e_5	8	v_4^5 ($t(v_4^5) = 3$)
(v_3, v_4)	e_6	6	v_5^6 ($t(v_5^6) = 1$)
(v_3, v_5)	e_7	14	v_1^7 ($t(v_1^7) = 8$), v_2^7 ($t(v_2^7) = 8$), v_4^7 ($t(v_4^7) = 9$)
(v_4, v_5)	e_8	10	v_1^8 ($t(v_1^8) = 9$), v_2^8 ($t(v_2^8) = 3$)

Figure 4.2 depicts the graphs of type-I distance functions $d(v_l, x)$, $x \in e_7$, for all l. We see that edge e_7 can be partitioned into linear segments $[v_3, v_2^7]$, $[v_2^7, v_4^7]$, and $[v_4^7, v_5]$.

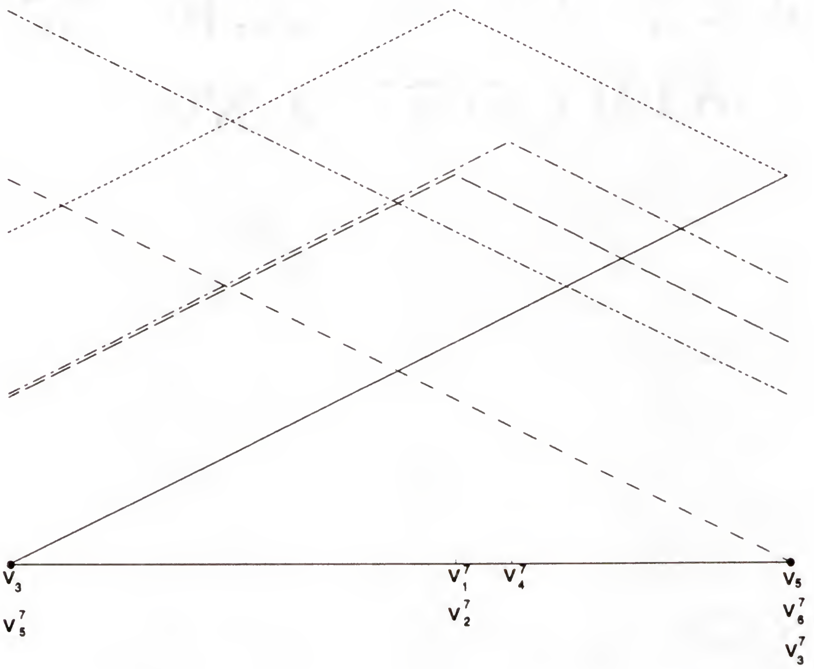


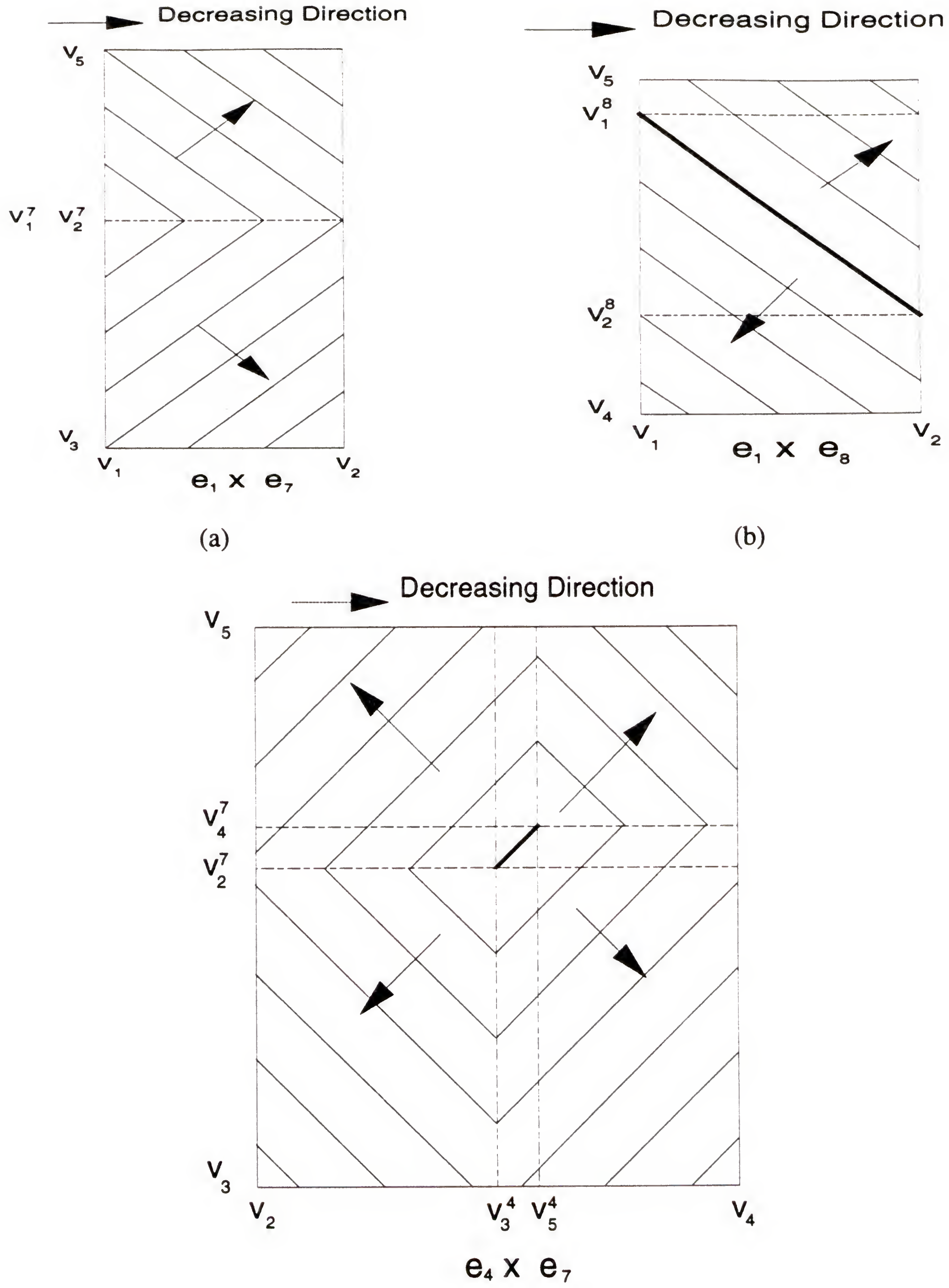
Figure 4.2 Examples of Linear Segments and Type-I Distances over an Edge for the Network of Figure 4.1

Example 4.2. Type-II distances have similar properties. Figures 4.3a to 4.3d depict the contour sets of $d(z_1, z_2)$ over domains $e_1 \times e_7$, $e_1 \times e_8$, $e_4 \times e_7$, and $e_4 \times e_8$. We see that $d(z_1, z_2)$ is piecewise linear and concave over its respective domains. Figure 4.3e depicts the contour set of $d(z_1, z_2)$ over $e_8 \times e_8$. In this case, $d(z_1, z_2)$ is PLC. For all these cases, we can partition each $e_p \times e_q$ into at most four regions over each of which $d(z_1, z_2)$ is linear. For example, we can partition $e_4 \times e_8$ into

$$LR_1 = \{(z_1, z_2) \mid z_1 \in e_4,\ z_2 \in e_8,\ t(z_1) - t(z_2) \geq t(v_4^4) - t(v_2^8)\} \text{ and}$$
$$LR_2 = \{(z_1, z_2) \mid z_1 \in e_4,\ z_2 \in e_8,\ t(z_1) - t(z_2) \leq t(v_4^4) - t(v_2^8)\}.$$

Over LR_1, $d(z_1, z_2) = 12 - t(z_1) + t(z_2)$, and over LR_2 $d(z_1, z_2) = 18 + t(z_1) - t(z_2)$.

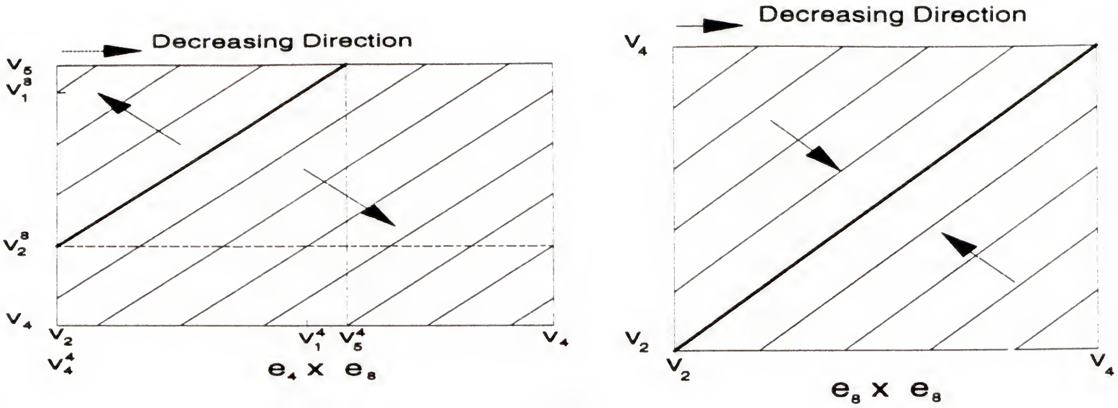
The fact that both types of distance functions are linear under proper restrictions motivates us to partition G^n into subsets such that over each subset all the distance functions are linear.



(a) (b)

(c)

Figure 4.3 Contours of $d(z_1, z_2)$



(d) (e)

Figure 4.3 Continued

Example 4.3. (Partition)

Consider a 2-facility problem defined on the network in Figure 4.1: P: Minimize $\{c(D(X)) \mid (x_1, x_2) \in e_4 \times e_8\}$, where $D(X) = (d(v_1, x_1) \ldots, d(v_5, x_1), d(v_1, x_2), \ldots, d(v_5, x_2), d(x_1, x_2))$. From Example 4.1, the linear segments in the two edges e_4 and e_8 are

$$e_4: L_1 = [v_2, v_1{}^4] \quad L_2 = [v_1{}^4, v_5{}^4] \quad L_3 = [v_5{}^4, v_4];$$
$$e_8: L_4 = [v_4, v_2{}^8] \quad L_5 = [v_2{}^8, v_1{}^8] \quad L_6 = [v_1{}^8, v_5].$$

The initial partition $e_4 \times e_8$ is $\{L_p \times L_q \mid p = 1, 2, 3, q = 4, 5, 6\}$. Over each $L_p \times L_q$, all the type-I distance functions are linear. From Figure 4.3d, $d(x_1, x_2)$ is nonlinear over $L_1 \times L_5$ and $L_2 \times L_6$. Also from Figure 4.3d, $L_1 \times L_5$ ($L_2 \times L_6$) can be further partitioned into two subsets $(L_1 \times L_5) \cap LR_1$ and $(L_1 \times L_5) \cap LR_2$ ($(L_2 \times L_6) \cap LR_1$ and $(L_2 \times L_6) \cap LR_2$), where LR_1 and LR_2 are given in Example 4.2. Over all these latter four subsets, $d(x_1, x_2)$ is linear. Thus, the final partition of $e_4 \times e_8$ is $\Omega = \{L_1 \times L_4, (L_1 \times L_5) \cap LR_1, (L_1 \times L_5) \cap LR_2, L_1 \times L_6, L_2 \times L_4, L_2 \times L_5, (L_2 \times L_6) \cap LR_1, (L_2 \times L_6) \cap LR_2, L_3 \times L_4, L_3 \times L_5, L_3 \times L_6\}$. For any $S \in \Omega$, subproblem Minimize $\{c(D(X)) \mid X \in S\}$ is a convex programming problem, since D(X) is linear, and S is linearly constrained.

It is computationally impossible to solve problems of nontrivial size by solving every subproblem. The next example shows the effect of implicit enumeration in reducing the number of subproblems actually solved and gives insight into what our PLC underestimates are.

Example 4.4. (Implicit Enumeration)

Consider a 2-multicenter problem defined on the network in Figure 4.1:

P: Minimize $\max\{f_1(x_1), f_2(x_2), 3d(x_1, x_2) \mid (x_1, x_2) \in G^{2'} = (e_1 \cup e_4) \times (e_7 \cup e_8)\}$,

where $f_j(x_j) = \max\{w_{ij}d(v_i, x_j), i = 1, \ldots, 5\}, j = 1, 2$, with w_{ij}'s listed below

i=	1	2	3	4	5	
w_{i1}	1	2	1	2	1	$v_{12} = 3$
w_{i2}	2	1	2	1	2.	

Suppose that, by some heuristic, we have an initial feasible solution X^0 to P with $x_1^0 = v_1$, $x_2^0 = v_4$, where $f(X^0) = 20$. Table 4.2 summarizes the implicit enumeration process.

Table 4.2 Subproblems and Their Lower Bounds

	LB	Solution Subsets	Improved Solution to P	Decomposition
P^1	26	$S^1 = e_1 \times e_7$		*Fathomed*
P^2	18.46	$S^2 = e_1 \times e_8$		P^5, P^6
P^3	26	$S^3 = e_4 \times e_7$		*Fathomed*
P^4	16	$S^4 = e_4 \times e_8$		P^7, P^8
P^5	19.2	$S^5 = \{e_1 \times e_8, t(x_1) + t(x_2) \leq 9\}$	$x_1^1 \in e_1$, $t(x_1^1) = 0$, $x_2^1 \in e_8$, $t(x_2^1) = 0.4$ $f(X^1) = 19.2$	*Fathomed*
P^6	26.7	$S^6 = \{e_1 \times e_8, t(x_1) + t(x_2) \geq 9\}$		*Fathomed*
P^7	16.5	$S^7 = \{e_4 \times e_8, t(x_1) - t(x_2) \geq -3\}$	$x_1^2 \in e_4$, $t(x_1^2) = 8.25$ $x_2^2 \in e_8$, $t(x_2^2) = 1.75$ $f(X^2) = 16.5$	 *Fathomed* *Optimal*
P^8	26.7	$S^8 = \{e_4 \times e_8, t(x_1) - t(x_2) \leq -3\}$		*Fathomed*

As in Table 4.2, we partition G' into subsets $e_1 \times e_7$, $e_1 \times e_8$, $e_4 \times e_7$, and $e_4 \times e_8$. The subproblems are $P^1, \ldots, P^4$. Figures 4.4a to 4.4d show, respectively, the graphs of $f_j(x_j)$, $j = 1, 2$, with x_j in the edges. Figures 4.3a to 4.3d showed the contour sets of $d(x_1, x_2)$ over the four subsets. Since $\min\{f_2(x_2) \mid x_2 \in e_7\} = 26 > f(X^0)$, subsets $e_1 \times e_7$ and $e_4 \times e_7$ are discarded. The lower bounding problem P_L^2 of P^2 is derived as follows: over e_1, $f_1(x_1) = \max\{14 - t(x_1), 12 + 2t(x_1)\}$ is PLC; over e_8, $f_2(x_2)$ is not convex, but we use the PLC supporting plane $pl(x_2) = \max\{20 - t(x_2),$

$1.5t(x_2) + 13\}$ as an underestimate. Function $d(x_1, x_2)$ over $e_1 \times e_8$ is not convex either. We use a PLC underestimate $d(x_1, x_2)^- = \max\{t(x_1) - 0.4t(x_2) + 6, -t(x_1) + 0.8t(x_2) + 6\}$. The first function in $d(x_1, x_2)^-$ is the linear plane in E^3 passing through points $A = (t(v_1), t(v_4), d(v_1, v_4))$, $B = (t(v_2), t(v_4), d(v_2, v_4))$, and $C = (t(v_2), t(v_5), d(v_2, v_5))$, and the second function in $d(x_1, x_2)^-$ is the linear plane in E^3 passing through points B, $D = (t(v_1), t(v_5), d(v_1, v_5))$, and C. Thus,

$P_L{}^2$: Minimize z

$$
\begin{aligned}
\text{s.t. } & \max\{14 - t(x_1), 12 + 2t(x_1)\} \le z \\
& \max\{20 - t(x_2), 1.5t(x_2) + 13\} \le z \\
& 3\max\{t(x_1) - 0.4t(x_2) + 6, -t(x_1) + 0.8t(x_2) + 6\} \le z \\
& 0 \le t(x_1) \le 6,\ 0 \le t(x_2) \le 10,\ z \ge 0.
\end{aligned}
$$

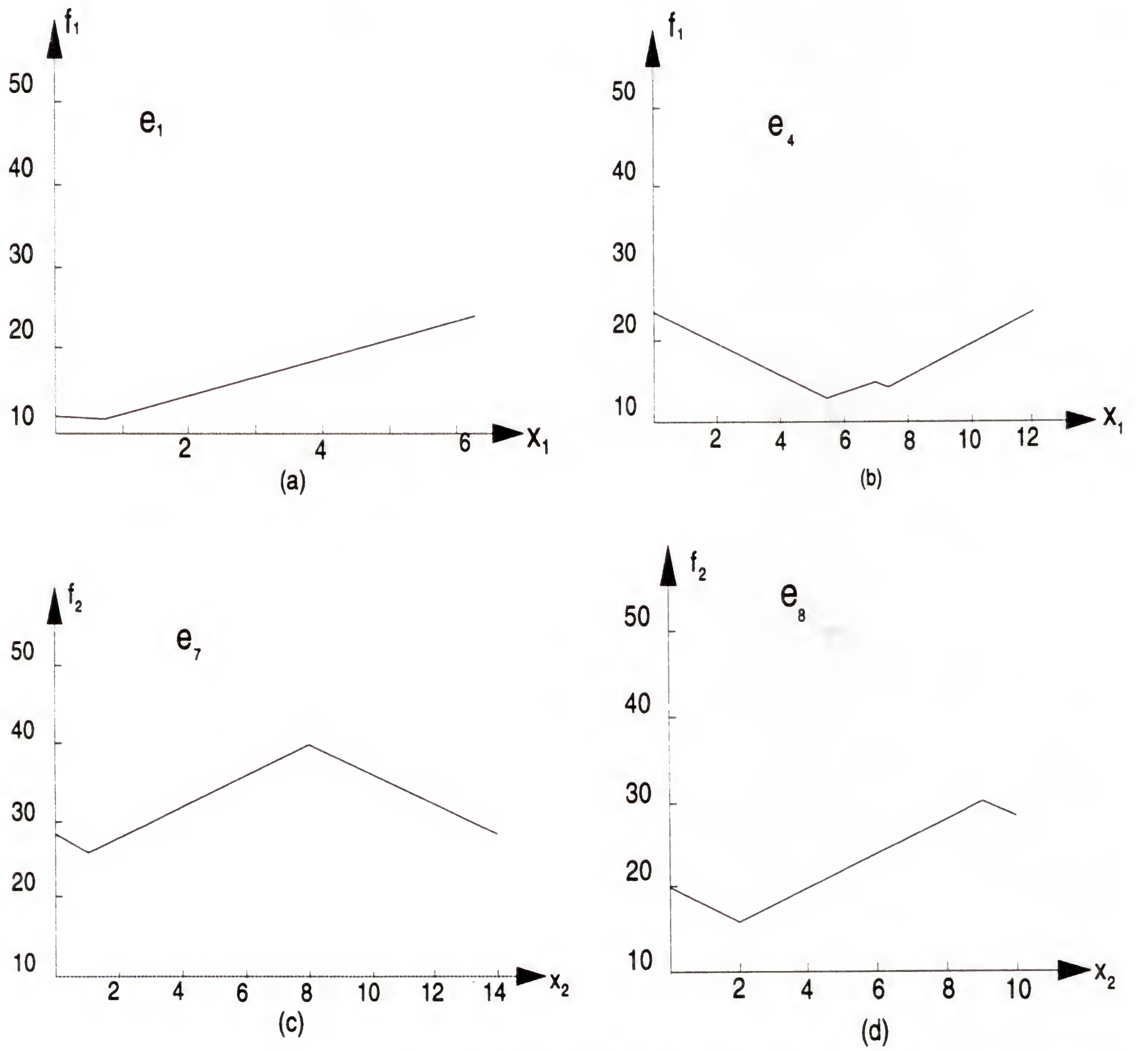


Figure 4.4 The Graphs on Edges

To decompose P^2, we partition $e_1 \times e_8$ to improve the approximation of $d(x_1, x_2)$. Set $e_1 \times e_8$ is further partitioned into S^5 and S^6 with hyperplane $t(x_1) + t(x_2) = 9$. Note that this hyperplane coincides with the line segment in $e_1 \times e_8$, on which $d(x_1, x_2)$ reaches its maximum inside $e_1 \times e_8$. Also note that $d(x_1, x_2)$ is linear over S^5 and S^6. Thus, the lower bounding problems for P^5 and P^6 are obtained by using the same PLC underestimates on $f_1(x_1)$ and $f_2(x_2)$ as in $P_L{}^2$.

After deriving similar PLC underestimates for $P_L{}^4$, we have

$P_L{}^4$: Minimize z

$$\begin{aligned}
\text{s.t. } & \max\{24 - t(x_1), 0.6667t(x_1) + 9.78, 2t(x_1)\} \le z \\
& \max\{20 - t(x_2), 1.5t(x_2) + 13\} \le z \\
& \max\{-t(x_1) - 0.4t(x_2) + 12, 0.1667t(x_1) + (x_2) - 2\} \le z \\
& 0 \le t(x_1) \le 12,\ 0 \le t(x_2) \le 10,\ z \ge 0.
\end{aligned}$$

We partition $e_4 \times e_8$ into LR_1 and LR_2 as given in Example 4.2. We have shown in Example 4.2 that $d(x_1, x_2) = 12 - t(x_1) + t(x_2)$ for any $(x_1, x_2) \in LR_1$ and $d(x_1, x_2) = 18 + t(x_1) - t(x_2)$, for any $(x_1, x_2) \in LR_2$.

From the above example, we see that for multifacility problems on general cyclic networks, the PLC lower bounding problems usually are not available without stringent conditions (e.g. each variable must be restricted to an edge). This is not the case for problems on grid networks.

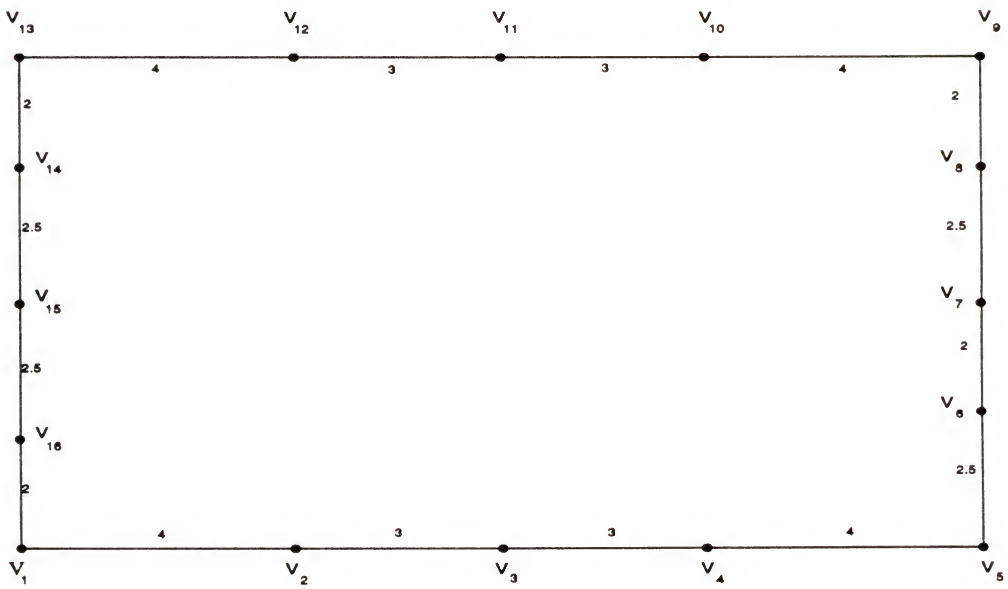


Figure 4.5 A Grid Network

Grid network distances have rectilinear distances as universal PLC underestimates. In case the rectilinear underestimates become inadequate, for example when a grid network is sparse, we can construct other PLC underestimates which have much better approximation and require less restriction. In the following several examples we emphasize the difference in the approximation quality between the rectilinear underestimates and the new PLC underestimates that we will formally introduce later in this chapter.

Example 4.5. Let N_g be the single cycle grid network shown in Figure 4.5 (The edge lengths are marked beside the edges in the figure). Let vl_l, vl_r, hl_b, and hl_t denote, respectively, the left vertical, the right vertical, the bottom, and the top horizontal grid lines (vl_l contains vertex v_{15}). A point on N_g has a coordinate (u_x, u_y) in E^2.

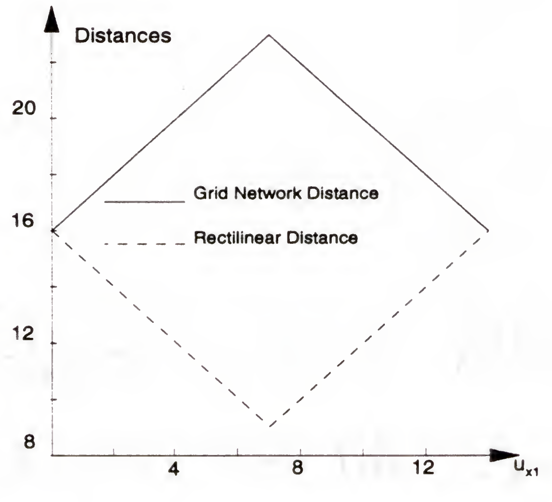


Figure 4.6 The Graphs of $d(v_3, u)$ and $r(v_3, u)$

Figure 4.6 depicts the graphs of $d(v_3, u)$ and the rectilinear distance $r(v_3, u)$ as u is restricted to hl_t. We see that over any interval [a, b] in hl_t, the linear supporting plane of $d(v_3, u)$ over that interval is a better underestimate than $r(v_3, u)$. As for the type-II distance, consider $d(u_1, u_2)$ and $r(u_1, u_2)$ as u_1 and u_2 are restricted to hl_b and hl_t respectively. Since u_{y1} and u_{y2} are fixed (at 0 and 9), $d(u_1, u_2)$ and $r(u_1, u_2)$ are functions of u_{x1} and u_{x2}, i.e. $d(u_1, u_2) = \delta_x(u_{x1}, u_{x2}) + 9$ and $r(u_1, u_2) = |u_{x1} - u_{x2}| + 9$. Figures 4.7a and 4.7b depict the contour sets of $\delta_x(u_{x1}, u_{x2})$ and $|u_{x1} - u_{x2}|$ respectively. Let $S = [lb_1, rb_1]\times[lb_2, rb_2]$, with $[lb_j, rb_j]$ an interval in E^1. We see that $|u_{x1} - u_{x2}|$

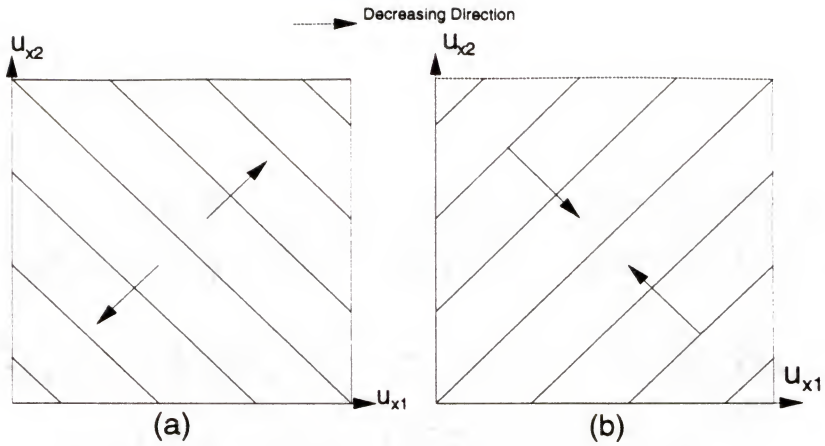


Figure 4.7 The Contours of $d(u_1, u_2)$ and $r(u_1, u_2)$

is not a good underestimate of $\delta_x(u_{x1}, u_{x2})$ if S is large or S is in the interior of $hl_b \times hl_t$. In this case, it is possible to construct a PLC underestimate of $\delta_x(u_{x1}, u_{x2})$ as follows. Let $pl_1(u_{x1}, u_{x2})$ be the linear function corresponding to the linear plane in E^3 passing through points $A = (lb_1, lb_2, \delta_x(lb_1, lb_2))$, $B = (rb_1, lb_2, \delta_x(rb_1, lb_2))$, $C = (rb_1, rb_2, \delta_x(rb_1, rb_2))$, and $pl_2(u_{x1}, u_{x2})$ be the linear function corresponding to the linear plane passing through points A, $D = (lb_1, rb_2, \delta_x(lb_1, rb_2))$, and C. Then, as we will show later in this chapter, $pl(u_{x1}, u_{x2}) = \max\{pl_1(u_{x1}, u_{x2}), pl_2(u_{x1}, u_{x2})\}$ is an underestimate of $\delta_x(u_{x1}, u_{x2})$. For example, with $lb_j = 4$, $rb_j = 10$, $j = 1, 2$, $pl(u_{x1}, u_{x2}) = \max\{u_{x1} - u_{x2} + 8, -u_{x1} + u_{x2} + 8\}$. The function $pl(u_{x1}, u_{x2})$ over $[4, 10]\times[4, 10]$ is considerably better than $|u_{x1} - u_{x2}|$, as shown in the following:

(u_{x1}, u_{x2})	(4, 4)	(10, 4)	(10, 10)	(4, 10)	(7, 7)	(8.5, 5.5)	(8.5, 8.5)
$d_x(u_{x1}, u_{x2})$	8	14	8	14	14	14	11
$pl(u_{x1}, u_{x2})$	8	14	8	14	8	11	8
$\|u_{x1} - u_{x2}\|$	0	6	0	6	0	3	0.

In the following two examples, we will show that under some circumstances, we can find much better PLC underestimates for the objective function.

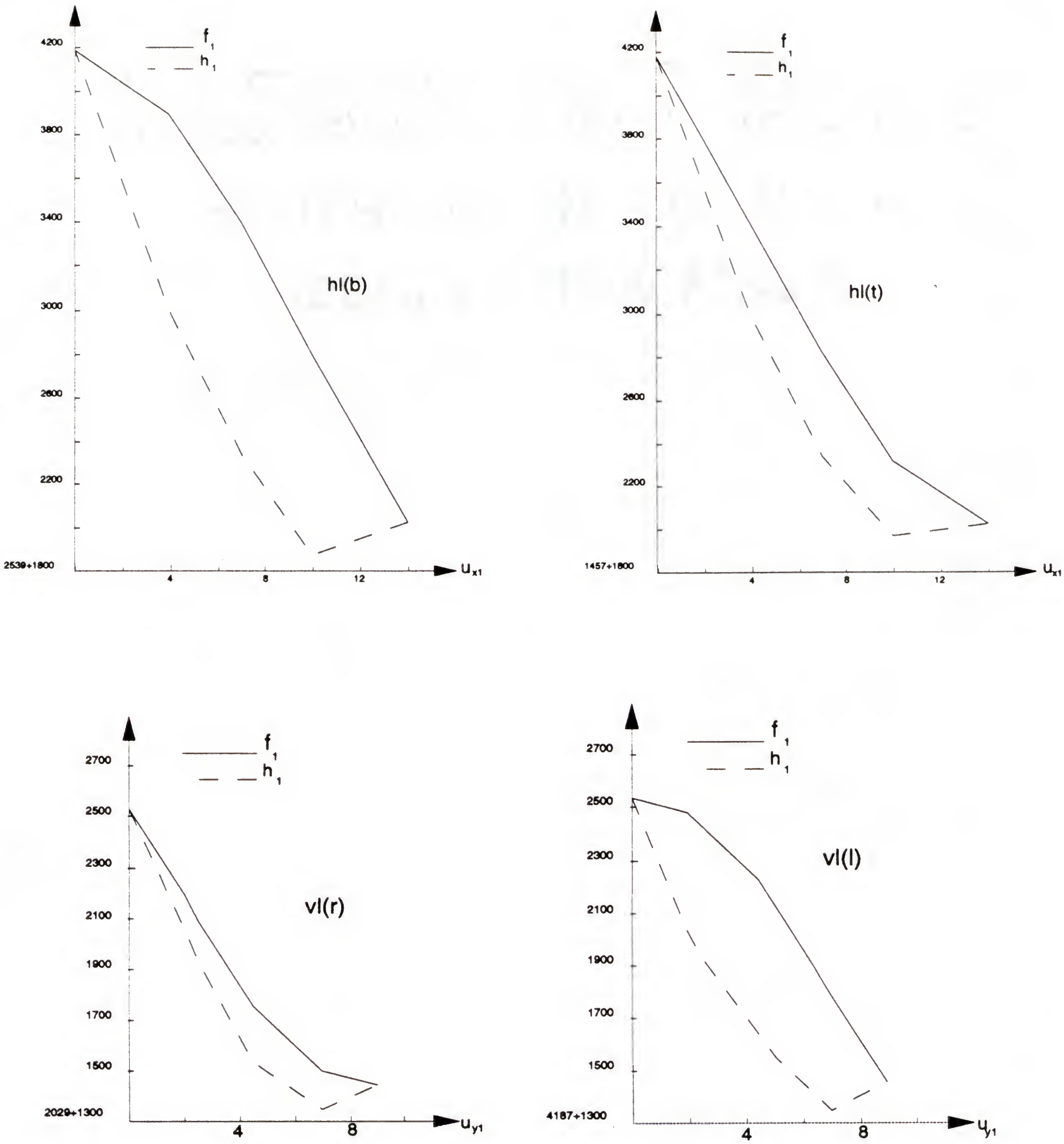


Figure 4.8 Distance Function Graphs

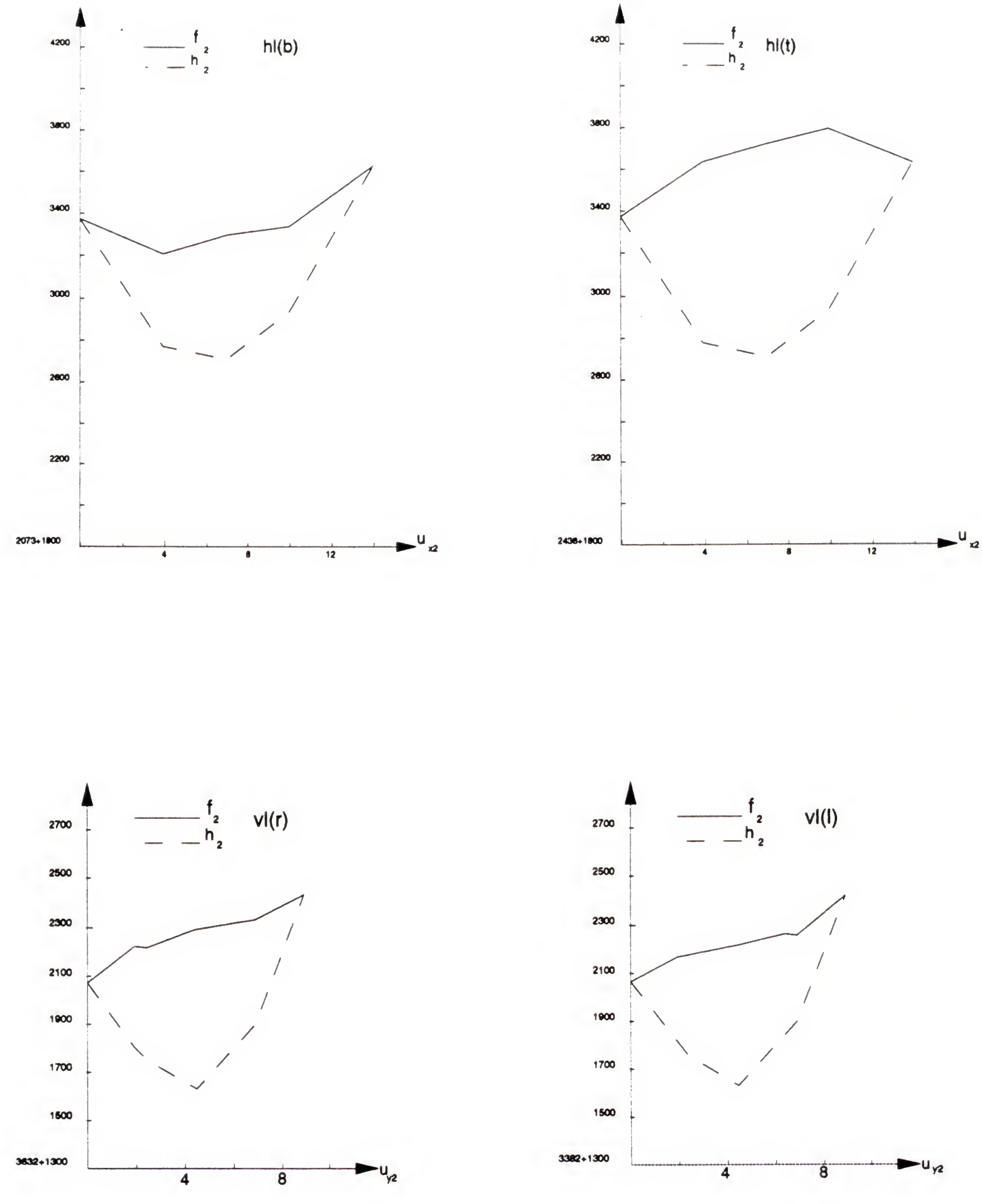


Figure 4.8 – Continued

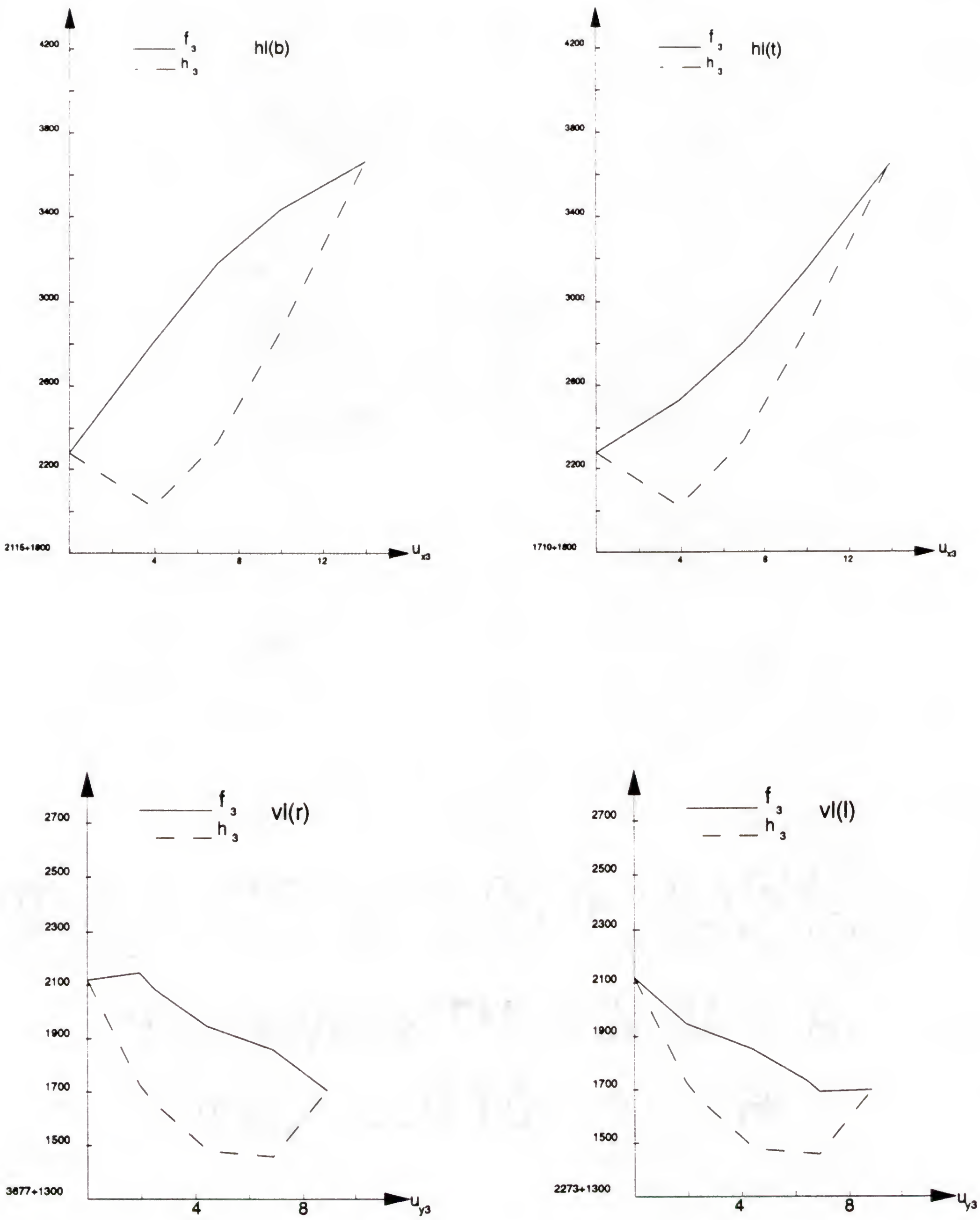


Figure 4.8 – Continued

Example 4.6. Let P be a multimedian problem: Minimize $\Sigma_j f_j(u_j) + f_{NN}(u_1, u_2, u_3)$ defined on the grid network N_g in Figure 4.5. The randomly generated weights are

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
w_{i1}	24	23	9	22	21	10	49	52	72	71	26	16	12	8	17	12	$v_{12} = 21, v_{13} = 10,$
w_{i2}	46	45	27	36	30	23	38	31	41	12	25	19	23	43	49	13	$v_{23} = 3$
w_{i3}	42	16	15	18	22	11	25	23	17	23	26	35	48	47	12	45	

Let $h_j(u_j)$ denote the function derived from $f_j(u_j)$ by replacing grid network distances with rectilinear distances. Figure 4.8 depicts the graphs of $f_j(u_j)$ and $h_j(u_j)$, j = 1, 2, and 3, as u_j moves along the grid lines. Notation hl(b), hl(t), vl(l), and vl(r) in the fiugres respectively indicate that the corresponding figures depict the graphs with u_j restricted to the bottom, the top, the left vertical, and the right vertical grid lines. We see that when u_j is restricted to a grid line, it is better to approximate $f_j(u_j)$ with its PLC supporting plane than to use $h_j(u_j)$. Also note that a PLC supporting plane approaches quickly to its original as the interval restriction gets smaller. In contrast, the function $h_j(u_j)$ does not approach $f_j(u_j)$. Through using, whenever possible, these PLC underestimates, in conjunction with the PLC underestimates of type-II distances discussed in Example 4.5 and the rectilinear underestimates, in a B&B process similar to the one discussed in Chapter 3, we solved P after examining 54 subproblems. If only use the rectilinear lower bounds, the B&B process needs to examine 256 subproblems before finding an optimal solution.

Example 4.7. Consider a 2-facility multicenter problem P: Minimize $f(u_1, u_2) = \max\{\max\{f_j(u_j), j = 1, 2\}\ f_{NN}(u_1, u_2)\}$ on the grid network N_g in Figure 4.9, with randomly generated weights

	1	2	3	4	5	6	7	8	9	10	11	12	
w_{i1}	6	6	3	3	6	6	7	9	7	8	3	9	$v_{12} = 20$
w_{i2}	9	6	8	2	5	10	3	7	9	3	6	6.	

Figure 4.10 depicts the graphs of $f_j(u_j)$ and $h_j(u_j)$ as u_j moves along the grid lines. We see that the PLC supporting planes of $f_j(u_j)$'s are better underestimates. Note that the graph of each $f_j(u_j)$ is either close to its PLC supporting plane, or consists of only a few linear segments. Thus, with some knowledge of the graphs, we can design proper branching heuristics to force the PLC supporting planes to quickly approach their originals. We used these underestimates, in conjunction with the rectilinear underestimates and the PLC underestimates of type-II distances,

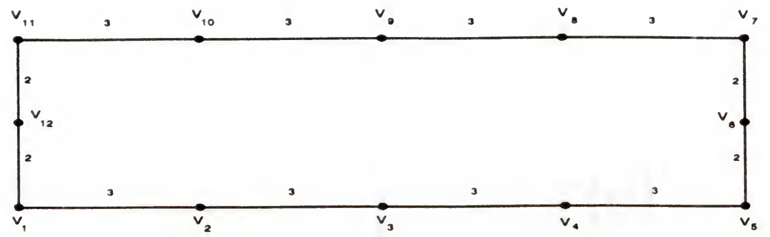


Figure 4.9 A Grid Network

to solve problem P after examining 18 subproblems. The lower bounding problems are all linear programs. With some preprocessing procedures discussed later in this chapter, we only needed to solve 12 linear programs, the largest of which had 10 constraints and 6 variables.

In summary, we introduced the notion of decomposing the solution set into subsets in which distance functions are linear. We demonstrated through examples the effectiveness of some PLC underestimates. The rest of this chapter is a formal exposition of the approaches shown in the examples.

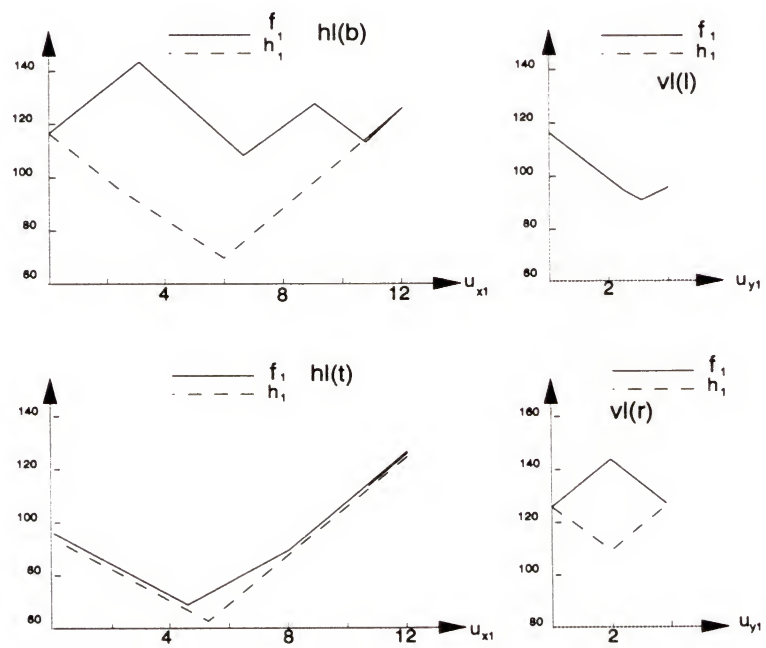


Figure 4.10 The Graphs of f and h

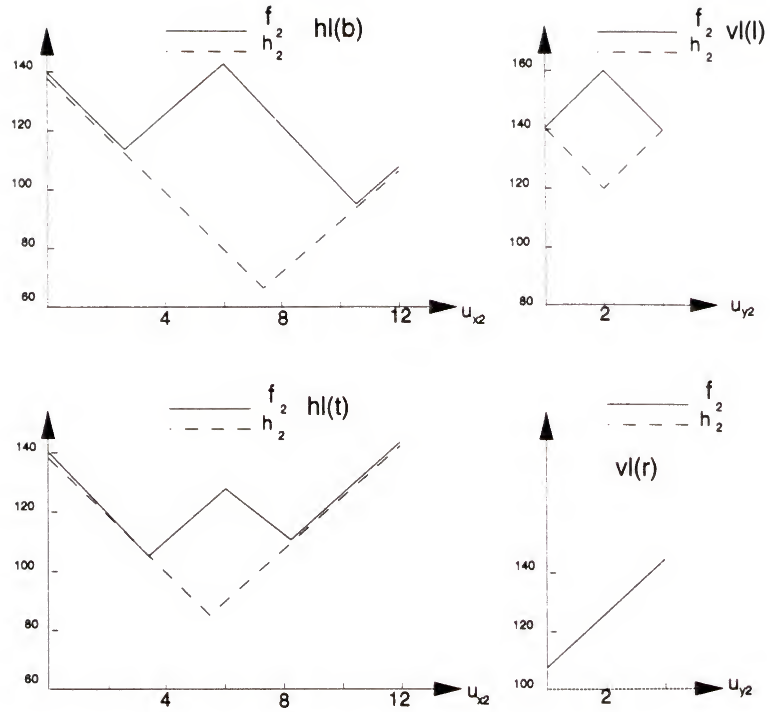


Figure 4.10 Continued

4.2 Review

Hooker (1986) used decomposition to solve a class of nonlinear single-facility network location problems. In Hooker (1989), he extended this approach to a class of nonlinear multifacility problems which involved only the type-I distance. Both works are based on the following observation.

Observation 4.1. (Hooker 1986)

Let (u, w) be an edge of G and let $s_0, s_1, \ldots s_k$ be the distinct antipodal points on the edge with $s_0 = u$ and $s_k = w$, $t(s_i) < t(s_{i+1})$ for i = 0, …, k−1. Then, in each linear segment $[s_i, s_{i+1}]$, i = 0, …, k−1, all the shortest distance functions d(v, x) are linear.

Hooker (1986) considered a nonlinear single facility network location problem P_s with an objective function of type-I distances -- $f(d(v_1, x), \ldots, d(v_n, x))$. He proposed decomposing P_s into subproblems each of which is defined on a linear segment. Since all the shortest distance functions are linear on a linear segment, each subproblem is a convex optimization problem if the function f is convex. A subgradient type lower bound for P_s on a given edge, and a domination relation among subproblems are given. An implicit enumeration algorithm is developed, which combines the lower bounding technique and the domination relation. Hooker (1989) extended this approach to the following nonlinear multifacility problem:

P_m: Minimize $\{f(d(v_1, X), \ldots, d(v_n, X)) \mid X \in G^n\}$

where f is a real-valued convex function on R^n and $d(v_i, X) = \min\{d(v_i, x_1), \ldots, d(v_n, x_n)\}$. Similar to the previous result, he proposed decomposing P_m into subproblems each of which is defined on a multi-set, which is an n-fold Cartesian product of n linear segments. To have an explicit $d(v_i, X)$, each vertex is assigned to a designated closest location variable; the multi-set is further decomposed into subsets for different vertex-variable assignments. A further attempt is made to reduce the number of subproblems actually solved by considering directional subgradients at every extreme point of an edge-set (a set in which every new facility variable is restricted to an edge).

4.3 An L-Partition of G^n and the Subproblems of P

In this section, we introduce a so-called L-partition Ω of G^n such that over each element of this partition both types of distances are linear. After defining Ω, we will define a specific type of subproblems of P that are useful in many implicit enumeration algorithms. Such a subproblem is defined through defining the corresponding solution subset.

4.3.1 The Type-II Distances

Hooker, Garfinkel, and Chen (1991), discussed the topology of a type-II distance function without giving any explicit form of the function. Since our results build upon the topology of type-II distances, we now discuss some properties of type-II distances.

Consider $d(z_1, z_2)$ over $e_p \times e_q$ for two arbitrary edges $e_p = (u_{[p]}, w_{[p]})$ and $e_q = (u_{[q]}, w_{[q]})$. A shortest path connecting z_1 and z_2 may contain end-points $u_{[]}$ and/or $w_{[]}$ depending on the locations z_1 and z_2 represent. Thus, $d(z_1, z_2)$ over $e_p \times e_q$ has the following expressions:

If p ≠ q then

$$d(z_1, z_2) = \min\left\{ \begin{array}{l} d(u_{[p]}, u_{[q]}) + t(z_1) + t(z_2), \\ d(w_{[p]}, u_{[q]}) + t(w_{[p]}) - t(z_1) + t(z_2), \\ d(u_{[p]}, w_{[q]}) + t(z_1) + t(w_{[q]}) - t(z_2), \\ d(w_{[p]}, w_{[q]}) + t(w_{[p]}) - t(z_1) + t(w_{[q]}) - t(z_2) \end{array} \right\} \qquad (4.2);$$

if e_p and e_q are the same edge, say (u, w), then

$$d(z_1, z_2) = \min\left\{ \begin{array}{l} d(u, w) + t(z_1) + t(w) - t(z_2) \\ |t(z_1) - t(z_2)| \\ d(u, w) + t(w) - t(z_1) + t(z_2) \end{array} \right\} \qquad (4.3);$$

in particular, if (u, w) is a shortest path between u and w, then (E4.3) becomes

$$d(z_1, z_2) = |t(z_1) - t(z_2)| \qquad (4.4).$$

For (4.2), $d(z_1, z_2)$ is the minimum of four linear functions, so that it is piecewise linear and concave. For (4.3), $d(z_1, z_2)$ is the minimum of 3 convex functions and is neither concave nor convex, but in its special case (4.4), it is PLC. We can express $d(z_1, z_2)$ explicitly as the following.

Let $u_{[q]}{}^p$ and $w_{[q]}{}^p$ ($w_{[p]}{}^q$, $u_{[p]}{}^q$) be the antipodal points of $u_{[q]}$ and $w_{[q]}$ ($u_{[p]}$ and $w_{[p]}$) on edge e_p (e_q) (From Hooker, Garfinkel, and Chen (1991), $t(w_{[q]}{}^p) \le t(u_{[q]}{}^p)$ if and only if $t(w_{[p]}{}^q) \le t(u_{[p]}{}^q)$ and $|t(u_{[q]}{}^p) - t(w_{[q]}{}^p)| = |t(u_{[p]}{}^q) - t(w_{[p]}{}^q)|$)

Case 1a. p ≠ q, $t(w_{[q]}{}^p) \le t(u_{[q]}{}^p)$ and $t(w_{[p]}{}^q) \le t(u_{[p]}{}^q)$.

$$d(z_1, z_2) = \begin{cases} d(u_{[p]}, u_{[q]}) + t(z_1) + t(z_2) & \text{If } t(z_1) \le t(u_{[q]}{}^p),\ t(z_2) \le t(u_{[p]}{}^q), \text{ and } t(z_1) + t(z_2) \le t(u_{[q]}{}^p) + t(w_{[p]}{}^q) \\ d(w_{[p]}, u_{[q]}) + t(w_{[p]}) - t(z_1) + t(z_2) & \text{If } t(z_1) \ge t(u_{[q]}{}^p) \text{ and } t(z_2) \le t(w_{[p]}{}^q) \\ d(u_{[p]}, w_{[q]}) + t(z_1) + t(w_{[q]}) - t(z_2) & \text{If } t(z_1) \le t(w_{[q]}{}^p) \text{ and } t(z_2) \ge t(u_{[p]}{}^q) \\ d(w_{[p]}, w_{[q]}) + t(w_{[p]}) - t(z_1) + t(w_{[q]}) - t(z_2) & \text{If } t(z_1) \ge t(w_{[q]}{}^p),\ t(z_2) \ge t(w_{[p]}{}^q), \text{ and } t(z_1) + t(z_2) \ge t(u_{[q]}{}^p) + t(w_{[p]}{}^q); \end{cases}$$

Case 1b. p ≠ q, $t(u_{[q]}{}^p) \le t(w_{[q]}{}^p)$ and $t(u_{[p]}{}^q) \le t(w_{[p]}{}^q)$

$$d(z_1, z_2) = \begin{cases} d(u_{[p]}, w_{[q]}) + t(z_1) + t(w_{[q]}) - t(z_2) & \text{If } t(z_1) \le t(w_{[q]}{}^p),\ t(u_{[p]}{}^q) \le t(z_2) \le t(w_{[q]}), \text{ and } t(z_1) - t(z_2) \le t(u_{[q]}{}^p) - t(u_{[p]}{}^q) \\ d(u_{[p]}, u_{[q]}) + t(z_1) + t(z_2) & \text{If } t(z_1) \le t(u_{[q]}{}^p) \text{ and } t(z_2) \le t(u_{[p]}{}^q) \\ d(w_{[p]}, w_{[q]}) + t(w_{[p]}) - t(z_1) + t(w_{[q]}) - t(z_2) & \text{If } t(w_{[q]}{}^p) \le t(z_1) \text{ and } t(w_{[p]}{}^q) \le t(z_2) \\ d(w_{[p]}, u_{[q]}) + t(w_{[p]}) - t(z_1) + t(z_2) & \text{If } t(u_{[q]}{}^p) \le t(z_1),\ t(z_2) \le t(w_{[p]}{}^q), \text{ and } t(z_1) - t(z_2) \ge t(u_{[q]}{}^p) - t(u_{[p]}{}^q); \end{cases}$$

Case 2a. $e_p = e_q = (u, w)$, and edge (u, w) is not a shortest path between u and w.

Let u' and w' be the antipodal points of u and w respectively on edge (u, w), we have

$$d(z_1, z_2) = \begin{cases} d(u, w) + t(z_1) + t(w) - t(z_2) & \text{If } t(z_1) \le t(w') \text{ and } t(z_2) \ge t(u') \\ |t(z_1) - t(z_2)| & \text{o/w} \\ d(u, w) + t(w) - t(z_1) + t(z_2) & \text{If } t(z_1) \ge t(u') \text{ and } t(z_2) \le t(w'); \end{cases}$$

Case 2b. $e_p = e_q = (u, w)$ and (u, w) is a shortest path between u and w.

$d(z_1, z_2) = |t(z_1) - t(z_2)|$.

As an example, the $d(z_1, z_2)$ with $z_1 \in e_4 = [v_2, v_4]$ and $z_2 \in e_7 = [v_3, v_5]$ in the network of Figure 4.1, has the following explicit form

$$d(z_1, z_2) = \begin{cases} d(v_2, v_5)+t(z_1)+t(v_5)-t(z_2) & t(z_1) \le t(v_5^4),\ t(v_2^7) \le t(z_2) \le t(v_5), \text{ and } t(z_1) - t(z_2) \le t(v_3^4) - t(v_2^7) \\ d(v_2, v_5)+t(z_1)+t(z_2) & t(z_1) \le t(v_3^4),\ t(z_2) \le t(v_2^4) \\ d(v_4, v_5)+t(v_4)-t(z_1)+t(v_4)-t(z_2) & t(v_5^4) \le t(z_1) \le t(v_4),\ t(v_2^7) \le t(z_2) \le t(v_5) \\ d(v_3, v_4)+t(v_4)-t(z_1)+t(z_2) & t(v_3^4) \le t(z_1) \le t(v_4),\ t(z_2) \le t(v_2^7), \text{ and } t(z_1) - t(z_2) \ge t(v_3^4) - t(v_2^7). \end{cases}$$

$$= \begin{cases} 18 + t(z_1) - t(z_2) & t(z_1) \le 7,\ 8 \le t(z_2) \le 18, \text{ and } t(z_1) - t(z_2) \le -2 \\ 6 + t(z_1) + t(z_2) & t(z_1) \le 6,\ t(z_2) \le 8 \\ 36 - t(z_1) - t(z_2) & 7 \le t(z_1) \le 12,\ 8 \le t(z_2) \le 14 \\ 18 - t(z_1) + t(z_2) & 6 \le t(z_1) \le 12,\ t(z_2) \le 8, \text{ and } t(z_1) - t(z_2) \ge -2. \end{cases}$$

Figure 4.11a and Figure 4.11b give the conceptual contour sets of $d(z_1, z_2)$ for Case 1a and Case 1b. The set of points at which $d(z_1, z_2)$ reaches its maximum over $e_p \times e_q$ form a line segment L_H. Points $(u_{[q]}^p, w_{[p]}^q)$ and $(w_{[q]}^p, u_{[p]}^q)$ are end points of this line segment and are, therefore, used to define this line segment.

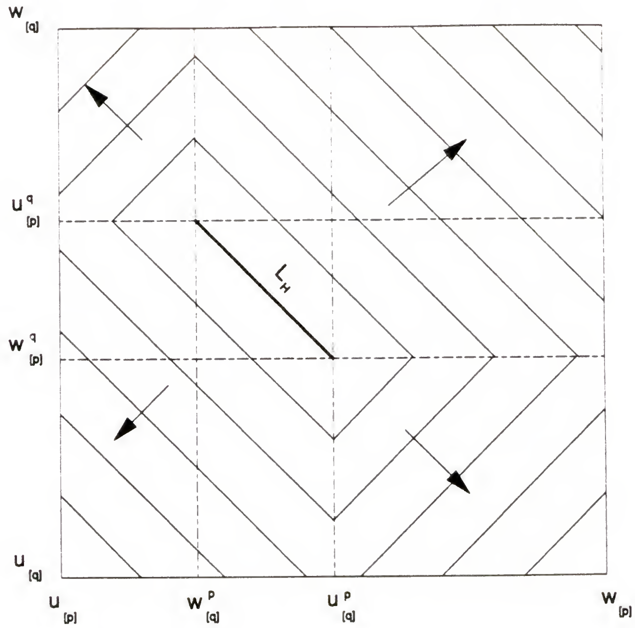
For Case 1a,

$L_H = \{(z_1, z_2) \in e_p \times e_q \mid t(z_1)+t(z_2) = t(u_{[q]}^p)+t(w_{[p]}^q),\ t(w_{[q]}^p) \le t(z_1) \le t(u_{[q]}^p),\ t(w_{[p]}^q) \le t(z_2) \le t(u_{[p]}^q)\}$,

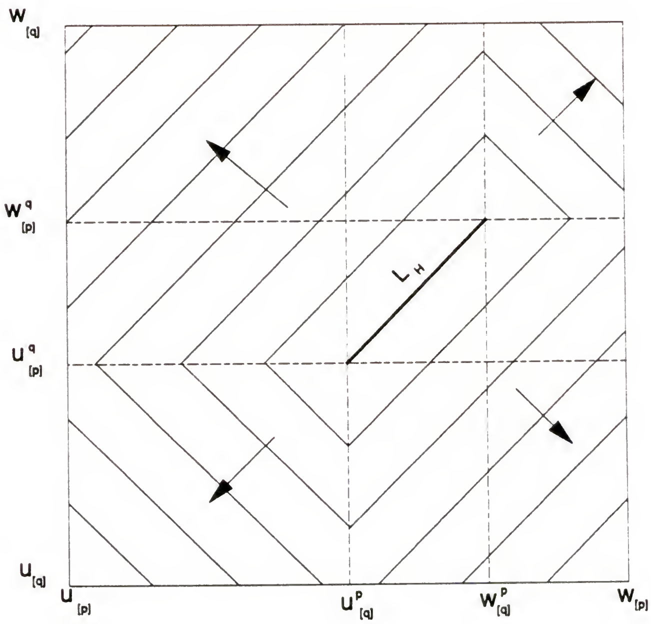
and for Case 1b,

$L_H = \{(z_1, z_2) \in e_p \times e_q \mid t(z_1)-t(z_2) = t(u_{[q]}^p)-t(u_{[p]}^q),\ t(u_{[q]}^p) \le t(z_1) \le t(w_{[q]}^p),\ t(u_{[p]}^q) \le t(z_2) \le t(w_{[p]}^q)\}$.

Figure 4.11c gives the conceptual contour set of $d(z_1, z_2)$ for Case 2b. The set of points at which $d(z_1, z_2)$ reaches its minimum over $e_p \times e_q$ form a line segment L_L. Points $(u_{[p]}, u_{[q]})$ and $(w_{[p]}, w_{[q]})$ are the two end points of L_L, so that $L_L = \{(z_1, z_2) \in e_p \times e_q \mid t(z_1) - t(z_2) = 0\}$.

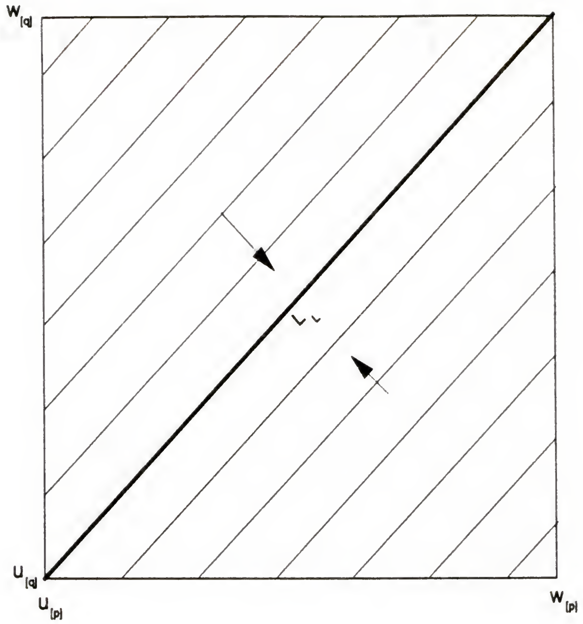


(a) Case 1a



(b) Case 1b

Figure 4.11 Contours of $d(z_1, z_2)$



(c) Case 2b
Figure 4.11 Continued

Finally, we make the following two assumptions about a cyclic network for the rest of this chapter. First, due to the following Property 4.0, we always assume that every edge in G is a shortest path between its two end points.

Property 4.0. If function c in P is non-decreasing, then there is an optimal solution to P with each new facility either on a vertex or in an edge which is a shortest path between its two end points.

Proof. See Appendix B.0.

Second, since $d(z_1, z_2)$ in Case 1a and Case 1b are symmetric, to simplify exposition, we assume that when z_1 and z_2 are restricted to two different edges, Case 1a is always true.

4.3.2 Linear Regions

Set $e_p \times e_q$ can be partitioned into subsets by some half-planes. When $e_p \neq e_q$, let H_{pq} be the hyperplane coinciding with the L_H in Figure 4.11a, so that $H_{pq} = \{(z_1, z_2) \in e_p \times e_q \,|\, t(z_1) + t(z_2) = t(u_{[q]}{}^p) + t(w_{[p]}{}^q)\}$. Let half-planes $H_{pq}^- = \{(z_1, z_2) \in e_p \times e_q | t(z_1) + t(z_2) \leq t(u_{[q]}{}^p) + t(w_{[p]}{}^q)\}$ and $H_{pq}^+ = \{(z_1, z_2) \in e_p \times e_q | t(z_1) + t(z_2) \geq t(z_{[q]}{}^p) + t(w_{[p]}{}^q)\}$. The $e_p \times e_q$ thus can be partitioned into

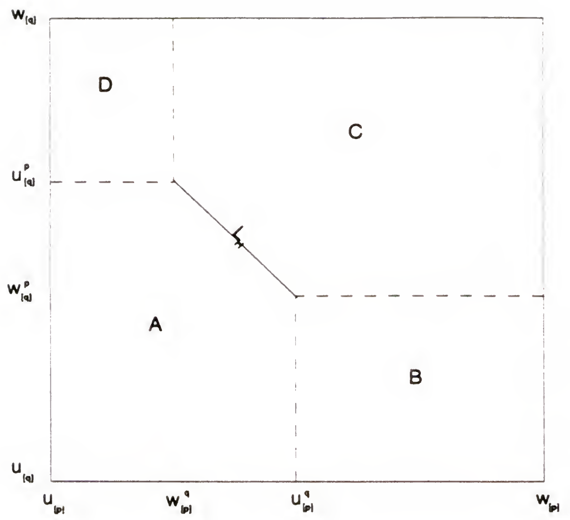


Figure 4.12 The Linear Regions

$$\{(z_1, z_2) \in e_p \times e_q \mid 0 \le t(z_1) \le t(u_{[q]}{}^p),\ 0 \le t(z_2) \le t(u_{[p]}{}^q)\} \cap H_{pq}{}^-,$$

$$\{(z_1, z_2) \in e_p \times e_q \mid t(u_{[q]}{}^p) \le t(z_1) \le t(w_{[j]}),\ 0 \le t(z_2) \le t(w_{[p]}{}^q)\},$$

$$\{(z_1, z_2) \in e_p \times e_q \mid t(w_{[q]}{}^p) \le t(z_1) \le t(w_{[p]}),\ t(w_{[p]}{}^q) \le t(z_2) \le t(w_{[q]})\} \cap H_{pq}{}^+,$$

and $$\{(z_1, z_2) \in e_p \times e_q \mid 0 \le t(z_1) \le t(w_{[q]}{}^p),\ t(u_{[p]}{}^q) \le t(z_2) \le t(w_{[q]})\}.$$

These four sets define, respectively, the regions A, B, C, and D shown in Figure 4.12.

If $e_p = e_q$, let $H_{pq} = \{(z_1, z_2) \in e_p \times e_q | t(z_1) - t(z_2) = 0\}$, $H_{pq}{}^+ = \{(z_1, z_2) \in e_p \times e_q | t(z_1) - t(z_2) \ge 0\}$ and $H_{pq}{}^- = \{(z_1, z_2) \in e_p \times e_q | t(z_1) - t(z_1) \le 0\}$. Here, hyperplane H_{pq} coincides with line segment L_L. The set $e_p \times e_q$ can be partitioned into $H_{pq}{}^+$ and $H_{pq}{}^-$, which correspond to the upper-left and the lower-right triangles in $e_p \times e_q$.

We call these regions the <u>linear regions</u>, since over each such region, $d(z_1, z_2)$ is linear.

<u>4.3.3 Partitioning G^n into Linear Sets</u>

With the linear segments for type-I distances and the linear regions for type-II distances, we can decompose G^n into finitely many <u>linear-sets</u> by combining the two structures. First, G^n is decomposed into <u>edge-sets</u> each of which has every new facility variable restricted to an edge.

For a given edge-set, let $e_{[j]}$ denote the edge to which x_j is restricted, let $\mathcal{L}_{[j]}$ denote a linear segment in $e_{[j]}$, and let $LR_{[j][k]}$ denote a linear region in set $e_{[j]} \times e_{[k]}$. A linear-set is

$$LS = \{X \in G^n | x_j \in \mathcal{L}_{[j]}, \text{ for all } j, (x_j, x_k) \in LR_{[j][k]} \text{ for all } j < k\}.$$

Then, the L-partition Ω of G^n is the set of all the nonempty linear-sets.

As for the number of elements in Ω, since in each edge there are at most m distinct antipodal points for a network of m vertices, there are at most (m−1) linear segments in each edge. There are at most $(|E|(m-1))^n$ different ways to assign x_j's to linear segments, so that there are $(|E|^n(m-1)^n)$ different type-I linear-sets, where a type-I linear-set is a subset of an edge-set, in which every x_j is restricted to a linear segment. For a type-I linear-set S, let NL be the set of location variable pairs (j, k) such that $d(x_j, x_k)$ is nonlinear over S. We see that in S, (x_j, x_k) is restricted to $\mathcal{L}_{[j]} \times \mathcal{L}_{[k]}$ where $\mathcal{L}_{[]}$ is the corresponding linear segment. If $d(x_j, x_k)$ is nonlinear, from Figure 4.11a and Figure 4.11c we see that $\mathcal{L}_{[j]} \times \mathcal{L}_{[k]}$ is shared by two linear regions of $e_{[j]} \times e_{[k]}$. Thus, we further decompose a type-I linear-set into at most $2^{|NL|}$ linear-sets by restricting each nonlinear new variable pair to one of the linear regions. All together, the total number of linear sets is $O(|E|^n(m-1)^n 2^{n(n+1)/2})$. On average, the number of linear sets is much less than this worst-case estimate because the number of distinct antipodal points in an edge should be much less than the worst-case and the number of nonlinear variable pairs in a given type-I linear-set is much less than the worst-case estimate n(n+1)/2.

For the multifacility problem P: Minimize $\{f(X) = c(D(X)) \mid X \in G^n\}$, let LS be a linear set $\{X \in G^n | x_j \in [s_{[j]}, s_{[j+1]}] \subseteq e_{[j]}$, for each j, and $(x_j, x_k) \in LR_{[j][k]}$ for every $j < k\}$, and let P' be the subproblem of P defined on LS. We can formulate P' as mathematical. programing problem. Over LS, any type-I distance $d(v_i, x_j)$ can be expressed as a linear function $\alpha_{ij} t(x_j) + \beta_{ij}$, where $\alpha_{ij} \in \{-1, 1\}$ and β_{ij} is a constant; any type-II distance function $d(x_j, x_k)$ can be expressed as a linear function $\rho_{jk} t(x_j) + \mu_{jk} t(x_k) + \eta_{jk}$, where $\rho_{jk}, \mu_{jk} \in \{-1, 1\}$, and η_{jk} is a constant. The constraints defining LS are the following: For each j, linear segment $[s_{[j]}, s_{[j]+1}]$ corresponds to constraints $t(s_{[j]}) \le t(x_j) \le t(s_{[j]+1})$; For each (j, k), j < k, linear region $LR_{[j][k]}$ corresponds to a constraint $a_{jk}{}^j t(x_j) + a_{jk}{}^k t(x_k) \le b_{jk}$, where $a_{jk}{}^j, a_{jk}{}^k \in \{-1, 1\}$. Thus

P': Minimize $c(\ \dots\ \alpha_{ij}t(x_j) + \beta_{ij}\ \dots,\ \rho_{jk}t(x_j) + \mu_{jk}t(x_k) + \eta_{jk},\ \dots)$

S. to $\quad a_{jk}{}^{j}t(x_j) + a_{jk}{}^{k}t(x_k) \le b_{jk}$ for each (j, k)

$\quad t(s_{[j]}) \le t(x_j) \le t(s_{[j]+1})$ for all j.

Problems P' is a linearly constrained convex optimization program. There may be many redundant constraints. We will consider removing redundant constraints in the next subsection.

4.3.4 The Subproblems of P

It is computationally impossible to solve P by solving all the subproblems each of which is defined on an element of Ω. More aggregated solution set paritions are necessary. In the following, we define a specific type of solution subset -- the L-sets. Each L-set is the union of some elements in Ω. We will also address how to represent and operate on these subsets.

Let H be the set of all the two-variable hyperplane H_{pq} (in those $e_p \times e_q$'s) for all $1 \le p \le q \le |E|$, and let H^- (H^+) be the sets of all the half-planes $H_{pq}{}^-$ ($H_{pq}{}^+$).

Observation 4.2. A linear set is defined by a set of single-variable half-planes each of which is of the form $\{x_j \in e_{[j]} \mid t(x_j) \le t(b)\}$, or $\{x_j \in e_{[j]} \mid t(x_j) \ge t(b)\}$ for some j and some antipodal point b; and a set of two-variable half-planes in $H^- \cup H^+$.

Now, we define an L-set. Note that Ω corresponds to the set of leaf nodes of a bi-decomposition tree (bi-tree), which has G^n as the root node and has each intermediate node decomposed into two nodes by a single-variable half-plane or a two-variable half-plane. Each intermediate node thus is the union of those elements in Ω, which are the leaf nodes of the subtree rooted by it. Also note that different orders of decomposition result in different bi-trees, and any two bi-trees have the same set of leaf nodes Ω, but their intermediate nodes are not all identical.

Since the local maxima of distance functions occur at antipodal points and at the hyperplanes in H, we thus assume that the branching strategy in an implicit enumeration algorithm is to partition a solution subset with either a single-variable hyperplane associated with some antipodal point, or a two-variable hyperplane in H. Two implicit enumeration algorithms thus differ only in the order of applying these hyperplanes. Thus, the branching tree generated by an implicit enumeration algorithm is a subtree of some bi-tree. In other words, every solution subset considered in an implicit enumeration algorithm is the union of some elements of Ω. We thus call these subsets L-sets. Formally, an L-set is defined as

Definition 4.1. An L-set S is a subset of G^n such that

$$S = \{X \in G^n \mid x_j \in CL_{[j]} \subseteq e_{[j]}, \text{ for each } j \in J', \text{ and } (x_j, x_k) \in K_{[j][k]} \text{ for } (j, k) \in B\},$$

where J' is a subset of index set J, $CL_{[j]}$ is a segment of some edge $e_{[j]}$ with two anti-podal points as boundaries, $B \subset \{(j, k) \mid j < k, j, k \in J'\}$, and $K_{[j][k]}$ is either the $H_{[j][k]}{}^+$ or the $H_{[j][k]}{}^-$ in $e_{[j]} \times e_{[k]}$.

Example 4.8. Consider the network G shown in Figure 4.1. Let P be a 3-facility problem. With $H_{18}{}^+ = \{(z_1, z_2) \in e_1 \times e_8 \mid t(z_1) + t(z_2) \geq 9\}$, the upper-right quadrilateral shown in Figure 4.3b, $S = \{(x_1, x_2, x_3) \mid x_1 \in e_1\ x_2 \in e_8, x_3 \text{ unrestricted}, (x_1, x_2) \in H_{18}{}^+\}$ is an L-set. Set S is the union of those linear sets of the following form:

$$LS = \{(x_1, x_2, x_3) \mid x_1 \in \mathcal{L}_{[1]} \subseteq e_1, x_2 \in \mathcal{L}_{[2]} \subseteq e_8, x_3 \in \mathcal{L}_{[3]} \subseteq e_{[3]}, (x_1, x_2) \in LR_{[1][2]} = H_{18}{}^+,$$
$$(x_1, x_3) \in LR_{[1][3]} \subseteq e_1 \times e_{[3]}, (x_2, x_3) \in LR_{[2][3]} \subseteq e_8 \times e_{[3]}\}.$$

Here, each $\mathcal{L}_{[]}$ is a linear segment and each $LR_{[][]}$ is a linear region.

The subproblems of P we are interested are the ones defined on L-sets.

Definition 4.2. An L-subproblem P' of P is a subproblem defined on an L-set S.

For the rest of this subsection, we study the structure of an L-set. An L-set is the intersection of a collection of single-variable half-planes and two-variable half-planes in E^n. It is desirable to represent an L-set with only its binding constraints. That is to represent an L-set S as the set $\{X \mid x_j \in L_{[j]}, \text{ for } j \in J', \text{ and } (x_j, x_k) \in R_{jk} \text{ for every } j < k\}$, where $L_{[j]} = \{x_j \mid X \in S\} \subseteq CL_{[j]}$ and $R_{jk} = \{(x_j, x_k) \mid X \in S\}$. This is important for controlling the number of constraints in lower bounding problems and improving PLC underestimates, since the quality of PLC underestimates is dependent on the sizes of those $L_{[j]}$'s and R_{jk}'s.

The simplicity of the constraints of L-set makes it possible to eliminate all its redundant constraints as the following. The set of binding constraints for an L-set satisfies

Constraint-Description 4.1:

There exists an index subset $J' \subseteq J$ such that

(a) for each $j \in J'$, x_j is restricted to an edge, say $e_{[j]}$,

(b) for each $j \in J'$, there are <u>at most</u> two single-variable half-planes which involve variable x_j,

(c) for each $j \in J'$, a single-variable half-plane involving x_j is of the form $\{x_j \in e_{[j]} \mid t(x_j) \leq \alpha_j\}$ or $\{x_j \in e_{[j]} \mid t(x_j) \geq \beta_j\}$ (If $\beta_j < \alpha_j$, then the linear-set is empty),

(d) for each (j, k), $j, k \in J'$, there is <u>at most</u> **one** two-variable half-plane involving both x_j and x_k, and such a half-plane must be $\{(x_j, x_k) \mid (x_j, x_k) \in K_{[j][k]}$, for some $K_{[j][k]} \in H^- \cup H^+\}$.

Since an L-set S (t(S) actually) is a polytope in E^n, we can build S by adding one constraint at a time. At each iteration, we detect and eliminate redundant constraints. It is thus sufficient to give an algorithm for the following problem:

<u>Adding a Constraint</u>: Let $S^0 \subset E^n$ be a nonempty set represented with binding constraints. Let S be derived from S^0 by imposing on S^0 either a single-variable half-plane or a <u>legitimate</u> two-variable half-plane (a legitimate two-variable half-plane involves two variables for which no other two-variable half-planes of S^0 involve both of them). Remove all the redundant constraints for S, or determine that S is empty.

In Appendix B.1, we give such an algorithm.

As for the geometry of $L_{[j]}$'s and R_{jk}'s, $L_{[j]}$ is a line segment; we know that for every $j, k \in J'$, either $R_{jk} = (L_{[j]} \times L_{[k]}) \cap K_{[j][k]}$ or $R_{jk} = L_{[j]} \times L_{[k]}$; $R_{jk} = G^2$ if j and k are not in J'; $R_{jk} = L_{[j]} \times G$ if $j \in J'$ and $k \notin J'$. For example, the L-set in Example 4.8 can be represented as

$$S = \{(x_1, x_2, x_3) \mid x_1 \in L_{[1]}, x_{[2]} \in L_2, (x_1, x_2) \in R_{12}\}$$

where $L_{[1]} = [v_1, v_2]$, $L_{[2]} = [v_2^8, v_5]$ (v_2^8 is the point in e_8 3 units distant from v_4 ($t(v_2^8) = 3$)), and $R_{12} = (L_{[1]} \times L_{[2]}) \cap H_{18}^+$.

<u>Observation 4.3</u>. Set R_{jk} defines either a triangle, a quadrilateral, or a pentagon in $e_{[j]} \times e_{[k]}$.

<u>Proof</u>. Either $R_{jk} = L_{[j]} \times L_{[k]}$, or $R_{jk} = (L_{[j]} \times L_{[k]}) \cap K_{[j][k]}$. For the first case, R_{jk} is a quadrilateral. For the second case, $K_{[j][k]}$ has a boundary $H_{[j][k]}$, the line segment running through $e_{[j]} \times e_{[k]}$ in 135 degrees (see Figure 4.11a, $H_{[j][k]}$ coincides with line segment L_H). Thus, the geometric shape of $R_{[j][k]}$ can only be a triangle, a quadrilateral, or a pentagon. ∎

4.4. Lower Bounding Problems

In this section, we discuss several lower bounding techniques for an L-subproblem P': Minimize $\{f(X) = c(D(X)) \mid X \in S\}$, where S is an L-set. First, under restriction S, we derive some PLC underestimates for distance functions, and then we combine these underestimates with the objectives of P to obtain lower bounding problems for P'.

4.4.1. PLC Underestimates for Both Types of Distances

Let $d(.,.)^-$ denote the PLC underestimate of $d(.,.)$. For completeness, we give every distance function an underestimate. We could only let $d(v_i, x_j)^- = 0$, and $d(x_j, x_k)^- = 0$, for x_j and/or x_k unrestricted (not in J'). The rest of this subsection is to find nontrivial PLC underestimates for those distances involving restricted variables.

4.4.1.1. The PLC Underestimate of $d(v_i, x_j)$ for $j \in J'$

Now, x_j is restricted to $L_{[j]} = [\alpha_j, \beta_j]$ – an edge-segment in some edge $e_{[j]}$. Since $d(v_i, x_j)$ is piecewise linear and concave over $L_{[j]}$, the best PLC underestimate for it is its linear supporting plane, which is the linear function running through points $(t(\alpha_j), d(v_i, \alpha_j))$ and $(t(\beta_j), d(v_i, \beta_j))$. As an example, consider $d(v_4, x_j)$ with x_j restricted to edge (v_3, v_5) of G_1 in Figure 4.1. In Figure 4.13, the dashed line is the linear supporting plane of $d(v_4, x_j)$ over $[v_3, v_5]$.

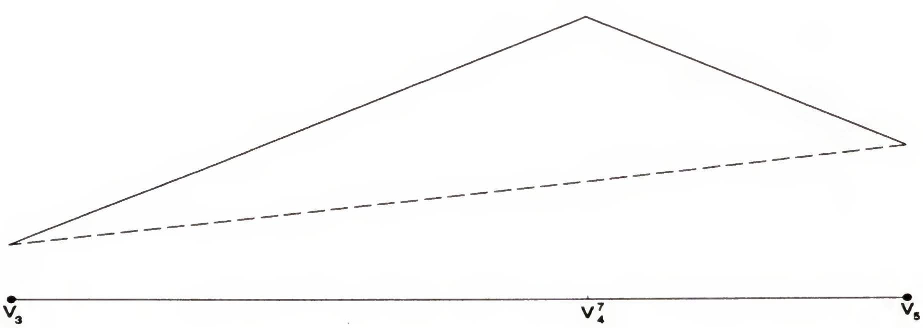


Figure 4.13 An Example Linear Support Plane

4.4.1.2. The PLC Underestimate of $d(x_j, x_k)$ for $j, k \in J'$

Now, $x_j \in L_{[j]} \subseteq e_{[j]}$, $x_k \in L_{[k]} \subseteq e_{[k]}$, and (x_j, x_k) is restricted to $R_{jk} = \{(x_j, x_k) \mid X \in S\}$.

Observation 4.4. If $e_{[j]} = e_{[k]}$, than $d(x_j, x_k)$ itself is PLC over $e_{[j]}\times e_{[k]}$. Therefore, $d(x_j, x_k)$ is PLC over R_{jk}, since R_{jk} is a subset of $e_{[j]}\times e_{[k]}$.

Observation 4.5. If R_{jk} is in a single linear region in $e_{[j]}\times e_{[k]}$, then $d(x_j, x_k)$ is linear over R_{jk}.

The following discussion then focuses on the case when $e_{[j]} \neq e_{[k]}$ and R_{jk} is not contained in a single linear region in $e_{[j]}\times e_{[k]}$. We construct $d(x_j, x_k)^-$ over R_{jk} as follows. We use p_j to denote an extreme point of R_{jk} and use P_j to denote the corresponding extreme point $(p_j, d(p_j))$ in $\{(t(\boldsymbol{x}), d(\boldsymbol{x})) \mid x \in R_{jk}, d(\boldsymbol{x})$ is the distance function value evaluated at $\boldsymbol{x}\}$. For R_{jk} a triangle with extreme points p_i, $i = 1, 2$, and 3, then let $d(x_j, x_k)^-$ be the function corresponding to the triangle in E^3 spanned by points $P_i = (p_i, d(p_i))$, $i = 1, 2$, and 3. Since $d(x_j, x_k)$ is concave over R_{jk}, $d(x_j, x_k)^-$ is an underestimate; Otherwise, define a quadrilateral super-set R_{jk}' of R_{jk} to be either R_{jk}, when R_{jk} is a quadrilateral, or $L_{[j]}\times L_{[k]}$, when R_{jk} is a pentagon. We actually use a $d(x_j, x_k)^-$ over R_{jk}' as the PLC underestimate of $d(x_j, x_k)$ over R_{jk}. Let p_1, p_2, p_3, and p_4 be the extreme points of R_{jk}' with p_1 and p_3 (p_2 and p_4) diagonal to each other.

Example 4.9. Consider network G in Figure 4.1 and an L-set $S = \{(x_1, x_2) \mid x_1 \in L_1 = [v_1, v_2], x_2 \in L_2 = [v_2^8, v_5]\}$. In this case, $R_{12} = L_1\times L_2 = R_{12}'$ is a quadrilateral, $p_1 = (v_1, v_2^8)$, $p_2 = (v_2, v_2^8)$, $p_3 = (v_2, v_5)$, $p_4 = (v_1, v_5)$, $P_1 = (t(v_1), t(v_2^8), d(v_1, v_2^8)) = (0, 3, 9)$, $P_2 = (t(v_2), t(v_2^8), d(v_2, v_2^8)) = (6, 3, 15)$, $P_3 = (t(v_2), t(v_5), d(v_2, v_5)) = (6, 10, 8)$, $P_4 = (t(v_1), t(v_5), d(v_1, v_5)) = (0, 10, 14)$.

Furthermore, let $\Delta(a, b, c)$ be the convex hull in E^p spanned by three linearly independent points a, b, and c in E^p. Note that R_{jk}' consists of $\Delta(p_2, p_1, p_4)$ and $\Delta(p_2, p_3, p_4)$ (R_{jk}' also consists of $\Delta(p_1, p_2, p_3)$ and $\Delta(p_1, p_4, p_3)$). In Figure 4.14b, we illustrate R_{jk}', p_i's, P_i's, and the triangles $\Delta(P_2, P_1, P_4)$, $\Delta(P_2, P_3, P_4)$, $\Delta(P_1, P_2, P_3)$ and $\Delta(P_1, P_4, P_3)$ (To help remembering, note that each triangle is uniquely associated with the extreme point in the middle position of the $\Delta(., ., .)$. For example, $\Delta(P_2, P_1, P_4)$ and $\Delta(P_2, P_3, P_4)$ differ in that one has point P_1 in the middle and the other has P_3 in the middle). Let $l_q(x_j, x_k)$, $q = 1, \ldots, 4$, be the algebraic forms of the linear planes

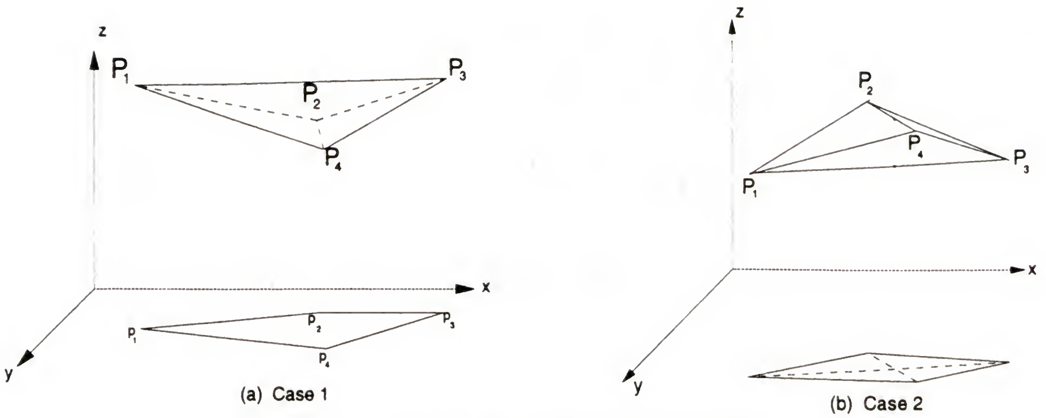


Figure 4.14 The Convex Hull

containing $\Delta(P_2, P_1, P_4)$, $\Delta(P_1, P_2, P_3)$, $\Delta(P_2, P_3, P_4)$, and $\Delta(P_1, P_4, P_3)$ respectively ($l_q(x_j, x_k)$ is the algebric form of triangle $\Delta(., P_q, .)$). Let s_{13} (s_{24}) be the 2-piecewise linear surface in E^3, formed by triangles $\Delta(P_2, P_1, P_4)$, $\Delta(P_2, P_3, P_4)$ ($\Delta(P_1, P_2, P_3)$ and $\Delta(P_1, P_4, P_3)$). Again, s_{pq} is the 2-piecewise linear surface consists of triangles $\Delta(., P_p, .)$ and $\Delta(., P_q, .)$. The algebraic representations of s_{13} and s_{24} are respectively

$$pl_1(x_j, x_k) = \begin{cases} l_1(x_j, x_k) & \text{if } (x_j, x_k) \in \Delta(p_2, p_1, p_4) \\ l_3(x_j, x_k) & \text{if } (x_j, x_k) \in \Delta(p_2, p_3, p_4), \end{cases}$$

and

$$pl_2(x_j, x_k) = \begin{cases} l_2(x_j, x_k) & \text{if } (x_j, x_k) \in \Delta(p_1, p_2, p_3) \\ l_4(x_j, x_k) & \text{if } (x_j, x_k) \in \Delta(p_1, p_4, p_3). \end{cases}$$

<u>Example 4.10</u>. For the R_{12}, R_{12}', p_i, P_i, $i = 1, \ldots, 4$, given in Example 4.9, we have

i	*triangles* in R_{12}'	Algebraic Representations
1	$\Delta(p_2, p_1, p_4)$	$\{(x_1, x_2) \mid x_1 \in L_1, x_2 \in L_2, 7t(x_1) + 6t(x_2) \leq 60\}$
2	$\Delta(p_2, p_3, p_4)$	$\{(x_1, x_2) \mid x_1 \in L_1, x_2 \in L_2, 7t(x_1) + 6t(x_2) \geq 60\}$
3	$\Delta(p_1, p_2, p_3)$	$\{(x_1, x_2) \mid x_1 \in L_1, x_2 \in L_2, -7t(x_1) + 3t(x_2) \leq 30\}$
4	$\Delta(p_1, p_4, p_3)$	$\{(x_1, x_2) \mid x_1 \in L_1, x_2 \in L_2, -7t(x_1) + 3t(x_2) \geq 30\}$

i	*triangles* in E^3	Algebraic Representations $l_i(x_1, x_2)$
1	$\Delta(P_2, P_1, P_4)$	$l_1(x_1, x_2) = t(x_1) + 0.7143t(x_2) + 6.8571$
2	$\Delta(P_2, P_3, P_4)$	$l_3(x_1, x_2) = -t(x_1) - t(x_2) + 24$
3	$\Delta(P_1, P_2, P_3)$	$l_2(x_1, x_2) = t(x_1) - t(x_2) + 12$
4	$\Delta(P_1, P_4, P_3)$	$l_4(x_1, x_2) = -t(x_1) + 0.7143t(x_2) + 6.8571.$

One can verify that $l_2(x_1, x_2) \geq l_4(x_1, x_2)$ when $(x_1, x_2) \in \Delta(p_1, p_2, p_3)$ and $l_4(x_1, x_2) \geq l_2(x_1, x_2)$ when $(x_1, x_2) \in \Delta(p_1, p_4, p_3)$. Therefore, from the definition of $pl_2(x_1, x_2)$, we have $pl_2(x_1, x_2) = \max\{l_2(x_1, x_2), l_4(x_1, x_2)\}$. This means surface s_{24} is PLC.

In general, one of the two surfaces s_{13} and s_{24} is always PLC. Figure 4.14a and 4.14b demonstrate, respectively, the cases when s_{13} and s_{24} are PLC.

Property 4.1. Over R_{jk}', either (a) $pl_1(x_j, x_k) = \max\{l_1(x_j, x_k), l_3(x_j, x_k)\}$, or (b) $pl_2(x_j, x_k) = \max\{l_2(x_j, x_k), l_4(x_j, x_k)\}$. If $P_1, \ldots, P_4$ are not in some linear plane, then exactly one of (a) and (b) is true.

Proof. In Appendix B.2, we give a geometric theory in E^3 of which Property 4.1 is a special case.

The following property states that in s_{13} and s_{24} the one which is PLC is the piecewise linear supporting plane of $d(x_j, x_k)$ over R_{jk}'.

Property 4.2. Let $pl(x_j, x_k) = pl_1(x_j, x_k)$ if (a) in Property 4.1 is true, and $pl(x_j, x_k) = pl_2(x_j, x_k)$ if (b) in Property 4.1 is true. Then, $pl(x_j, x_k)$ is a PLC underestimate for $d(x_j, x_k)$ over R_{jk}'.

Proof. Function $pl(.\,,.)$ is PLC by its definition. Without loss of generality, suppose $pl(x_j, x_k) = pl_1(x_j, x_k)$. For any point $(x_j, x_k) \in R_{jk}'$, since $\Delta(p_2, p_1, p_4) \cup \Delta(p_2, p_3, p_4) = R_{jk}'$, (x_j, x_k) is either in $\Delta(p_2, p_1, p_4)$ or $\Delta(p_2, p_3, p_4)$. Without loss of generality, suppose $(x_j, x_k) \in \Delta(p_2, p_1, p_4)$, so that point $(t(x_j), t(x_k), pl_1(x_j, x_k))$ is on $\Delta(P_2, P_1, P_4)$. Thus, we know that $(t(x_j), t(x_k), pl_1(x_j, x_k)) = \lambda_1 P_1 + \lambda_2 P_2 + \lambda_3 P_3$ for some $\lambda_1, \lambda_2, \lambda_3 \geq 0$, $\Sigma\lambda_i = 1$. Since the function $d(.\,,.)$ is concave, we know that $pl_1(x_j, x_k) = \lambda_1 d(p_1) + \lambda_2 d(p_2) + \lambda_3 d(p_3) \leq d(\lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3) = d(x_j, x_k)$. ∎

Now, we can give a procedure to construct $d(x_j, x_k)^-$.

Procedure 4.1. (Constructing $d(x_j, x_k)^-$ over R_{jk})

If R_{jk} is a triangle, then let $d(x_j, x_k)^-$ be the function corresponding to the triangle in E^3 spanned by points $P_1 = (t(p_1), d(p_1))$, $P_2 = (t(p_2), d(p_2))$, $P_3 = (t(p_3), d(p_3))$ where p_1, p_2, p_3 are the three extreme points of R_{jk}; If $e_{[j]} = e_{[k]}$ or R_{jk} is contained in one of the linear regions in $e_{[j]} \times e_{[k]}$, then let $d(x_j, x_k)^- = d(x_j, x_k)$; Otherwise, let $d(x_j, x_k)^-$ be the function $pl(x_j, x_k)$ defined in Property 4.2.

Example 4.11. Let P be on the G in Figure 4.1, P' a subproblem of P with $(x_j, x_k) \in L_{[j]} \times L_{[k]} = [v_3, v_5] \times [v_2, v_5^4] \subset e_7 \times e_4$. Thus, $d(x_j, x_k)$ is nonlinear over $L_{[j]} \times L_{[k]}$ (see Figure 4.3c). From Procedure 4.1, $R_{jk}' = L_{[j]} \times L_{[k]}$, and its extreme points are $p_1 = (v_3, v_2)$, $p_2 = (v_5, v_2)$, $p_3 = (v_5, v_5^4)$, and $p_4 = (v_3, v_5^4)$. The corresponding P-points in E^3 are $P_1 = (t(v_3), t(v_2), d(v_3, v_2)) = (0, 0, 6)$, P_2

$= (t(v_5), t(v_2), d(v_5, v_2)) = (14, 0, 8)$, $P_3 = (t(v_5), t(v_5^4), d(v_5, v_5^4)) = (14, 7, 16)$, $P_4 = (t(v_3), t(v_5^4), d(v_3, v_5^4)) = (0, 7, 12)$. The representations of $\Delta(P_2, P_1, P_4)$ and $\Delta(P_2, P_3, P_4)$ are, respectively, $l_1(x_j, x_k) = 0.1429t(x_j) + 0.8571t(x_k) + 6$ and $l_3(x_j, x_k) = 0.2857t(x_j) + 1.1429t(x_k) + 4$. One can

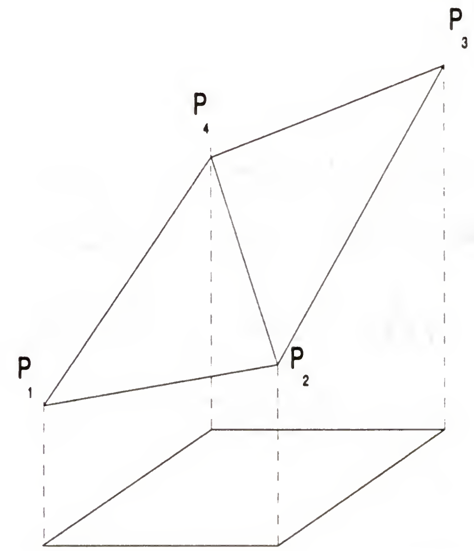


Figure 4.15 An Example of a PLC Underestimate

check that $l_3(x_j, x_k) \le l_1(x_j, x_k)$, $\forall(x_j, x_k) \in \Delta(p_2, p_1, p_4)$ and $l_1(x_j, x_k) \le l_3(x_j, x_k)$, $\forall(x_j, x_k) \in \Delta(p_2, p_3, p_4)$. Thus, the 2-piecewise linear surface s_{13}, which consists of triangles $\Delta(P_2, P_1, P_4)$ and $\Delta(P_2, P_3, P_4)$, is $pl_1(x_j, x_k) = \max\{l_1(x_j, x_k), l_3(x_j, x_k)\}$. Figure 4.15 illustrates the surface s_{13}.

4.4.2. A Lower Bounding Problem Based on Subgradients

In this subsection, we extend the subgradient lower bounding techniques suggested in Hooker (1986, 1989) to problems which involve both types of distances. The subproblem is P': Minimize $\{f(X) = c(D(X)) \mid X \in S\}$ where S is an L-set.

Let S_E be the set of all the extreme points of S.

Lemma 4.1. Let $X^E = (x_1^E, \ldots, x_n^E)$ be an extreme point in S_E. Let μ_{ij} denote the argument in function c corresponding to $d(v_i, x_j)$ and let τ_{jk} denote that corresponding to $d(x_j, x_k)$. Let $\nabla c(D(X^E)) = (\ldots, \theta_{ij}, \ldots, \xi_{jk}, \ldots)$ be a subgradient of c evaluated at $D(X^E)$ ($\theta_{ij} = \partial c/\partial\mu_{ij}$ and $\xi_{jk} = \partial c/\partial\tau_{jk}$ evaluated at $D(X^E)$, if c is differentiable). Then,

$$f(X) \ge C(X^E) + \Sigma_{i,j}\theta_{ij}d(v_i, x_j) + \Sigma_{j,k}\xi_{jk}d(x_j, x_k) \qquad \text{for any } X \in S \qquad (4.5)$$

where $C(X^E) = f(X^E) - \Sigma_{i,j}\theta_{ij}d(v_i, x_j^E) - \Sigma_{j,k}\xi_{jk}d(x_j^E, x_k^E)$.

Proof. Since c is convex, thus $f(X) \geq f(X^E) + \Sigma_{i,j}\theta_{ij}[d(v_i, x_j) - d(v_i, x_j^E)] + \Sigma_{j,k}\xi_{jk}[d(x_j, x_k) - d(x_j^E, x_k^E)]$ for any $X \in S$. With $C(X^E)$ the sum of all the constant terms, we have (4.5). ∎

Lemma 4.2. For any given extreme point $X^E \in S_E$ and a subgradient $(\ldots, \theta_{ij}, \ldots, \xi_{jk}, \ldots)$ of c evaluated at $D(X^E)$, we have

$$f(X) \geq C(X^E) + \Sigma_{i,j}\theta_{ij}d(v_i, x_j)^- + \Sigma_{j,k}\xi_{jk}d(x_j, x_k)^- \quad \text{for any } X \in S \quad (4.6).$$

Property 4.3. Let S' be a subset of S_E. The following problem is a lower bounding problem of P'.

P_{L1}': Minimize z

$$\begin{aligned} \text{s. t.} \quad & z \geq C(X^E) + \Sigma_{i,j}\,\theta_{ij}d(v_i, x_j)^- + \Sigma_{j,k}\xi_{jk}d(x_j, x_k)^- \quad \text{for } X^E \in S' \\ & X \in S. \end{aligned}$$

Since each $d(.\,,.)^-$ is PLC with at most two linear pieces, the above problem can be easily transformed into a linear programming problem. Since we are mostly interested in the multimedian and the multicenter problems, we now give their respective forms of (4.5) below.

Remark 4.1. For the multimedian problem, the subgradient lower bound in (4.5) is f(X) itself.

Remark 4.2. For the multicenter problem, a subgradient of c evaluated at some point D(X) for some $X \in G^n$ is a vector $(\ldots\, \theta_{ij} \ldots, \xi_{jk}, \ldots)$ where

$$\theta_{ij} = \begin{cases} 0 & \text{if } (i, j) \notin A(X) \\ w_{ij}\lambda_{ij} & \text{if } (i, j) \in A(X) \end{cases} \quad \text{and } \xi_{jk} = \begin{cases} 0 & \text{if } (j, k) \notin B(X) \\ v_{jk}\lambda_{jk} & \text{if } (j, k) \in B(X), \end{cases} \quad (4.7)$$

with A(X) and B(X) the sets of variable pairs with the corresponding weighted distance equal to f(X) (i.e. $A(X) = \{(i, j) \mid w_{ij}d(v_i, x_j) = c(D(X))\}$ and $B(X) = \{(j, k) \mid v_{jk}d(x_j, x_k) = c(D(X))\}$), and $\Sigma\{\lambda_{ij} \mid (i, j) \in A(X) \cup B(X)\} = 1$.

Proof. We can express c as $\max\{\alpha_1 y_1, \ldots, \alpha_p y_p\}$ where p is the number of arguments in c. For a given point $Y' = (y_1', \ldots y_p')$, let S(Y') denote the set of subgradients of c evaluated at Y'. We know that S(Y') is a convex set spanned by its extreme points. An extreme point of S(Y') is some vector $(0, \ldots, 0, \alpha_h, 0, \ldots, 0)$ such that $\alpha_h y_h' = \max\{\alpha_1 y_1', \ldots, \alpha_p y_p'\}$. Substituting α_h and y_h' with the corresponding weight and distance, an extreme point of S(D(X)) is either a vector $(0, \ldots, 0, w_{ij}, 0, \ldots 0)$ where $w_{ij}d(v_i, x_j) = c(D(X))$ or a vector $(0, \ldots, 0, v_{jk}, 0, \ldots, 0)$ where $v_{jk}\, d(x_j, x_k) =$

c(D(X)). Thus, as a convex combination of these extreme points of S(D(X)), a subgradient of c evaluated at D(X) is a vector $(\ldots\ \theta_{ij}\ \ldots,\ \xi_{jk},\ \ldots)$ as given in (4.7). ∎

Remark 4.3. For the multicenter problem, let A(X) and B(X) be the sets defined in Remark 4.2 for any $X \in G^n$. Then, for any given $X' \in S$,

$$f(X) \geq \Sigma_{(i,j)\in A(X')}\lambda_{ij}w_{ij}d(v_i, x_j) + \Sigma_{(j,k)\in B(X')}\lambda_{jk}v_{jk}d(x_j, x_k), \quad \text{for any } X \in S \qquad (4.8).$$

Proof. Replace the θ_{ij}'s and ξ_{jk}'s in (4.5) with the corresponding right-hand sides in (4.7) to get the conclusion. ∎

4.4.3. The Lower Bounding Problem Based on Distance Underestimates

For the subproblem P', we can construct a lower bounding problem P_{L2}' of P' by directly replacing each $d(v_i, x_j)$ $(d(x_j, x_k))$ with its underestimate $d(v_i, x_j)^-$ $(d(x_j, x_k)^-)$.

$$P_{L2}': \underset{X \in S}{\text{Minimize}}\ f(X)^- = c(\ \ldots,\ d(v_i, x_j)^-,\ \ldots,\ d(x_j, x_k)^-,\ \ldots).$$

Property 4.4. The optimal value of P_{L2}' is greater than or equal to that of P_{L1}' in Property 4.3.

Proof. The right-hand-side of (4.6) is a lower bound linear approximation of $c(\ldots,\ d(v_i, x_j)^-,\ \ldots,\ d(x_j, x_k)^-,\ \ldots)$. ∎

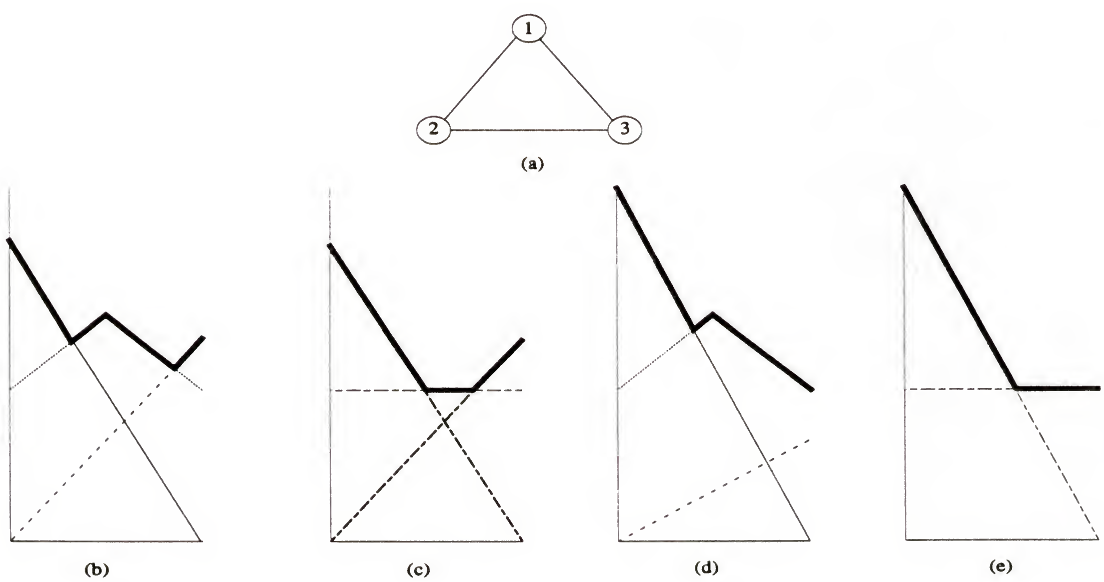


Figure 4.16 The Graphs of f_j and f_j^-

Example 4.12. Consider a multicenter problem on the equilateral triangle G in Figure 4.16(a).

P': $\underset{X \in S}{\text{minimize}}\ f(x_1, x_2) = \max\{\ 4d(v_1, x_1), 6d(v_2, x_1), 3d(v_3, x_1), 3d(v_1, x_2), 2d(v_2, x_2), 7d(v_3, x_2), 3.5d(x_1, x_2)\}$.

Here $S = [v_1, v_2]\times[v_2, v_3]$. The optimal value of P' is 3.43. We construct the subgradient-type lower bounding problem with the four extreme points in S, $X^{E1} = (v_1, v_2)$, $X^{E2} = (v_1, v_3)$, $X^{E3} = (v_2, v_2)$, and $X^{E4} = (v_2, v_3)$. The extremal subgradients evaluated at these extreme points are respectively $\nabla c(D(X^{E1})) = (0, 0, 0, 0, 0, 7, 0)$, $\nabla c(D(X^{E2})) = (0, 6, 0, 0, 0, 0, 0)$, $\nabla c(D(X^{E3})) = (0, 0, 0, 0, 0, 7, 0)$, $\nabla c(D(X^{E4})) = (4, 0, 0, 0, 0, 0, 0)$. According to Property 4.3, the subgradient-type lower bounding problem is

P_{L1}': Minimize $\{z \mid 4d(v_1, x_1) \le z, 6d(v_2, x_1) \le z, 7d(v_3, x_2) \le z, z \ge 0, (x_1, x_2) \in S\}$.

The optimal value of this lower bounding problem is 2.4. The optimal value of P' is 3.43.

Now, we consider the lower bounding problem obtained by substituting distance underestimates directly into function c. Figure 4.16(b) and (d) show the graphs of $d(v_i, x_1)$'s and $d(v_i, x_2)$'s respectively. The dark lines are the graphs of $f_1(x_1) = \max\{w_{i1}d(v_i, x_1), i = 1, 2, 3\}$ and $f_2(x_2) = \max\{w_{i2}d(v_i, x_2), i = 1, 2, 3\}$, respectively. According to the discussion in Subsection 4.3.2, we have $d(v_3, x_1)^- = 1$, $d(v_1, x_2)^- = 1$, $d(v_i, x_j)^- = d(v_i, x_j)$ for the rest of the (i, j)'s, and $d(x_1, x_2)^- = \max\{1 - t(x_1), t(x_2)\}$. Thus, the lower bounding problem is

P_{L2}': $\underset{X \in S}{\text{minimize}}\ f(x_1, x_2)^- = \max\{4d(v_1, x_1), 6d(v_2, x_1), 3, 3, 2d(v_2, x_2), 7d(v_3, x_2), 3.5d(x_1, x_2)^-\}$.

Figure 4.16(c) and 4.16(e) show the graph of each $d(v_i, x_1)^-$ and $d(v_i, x_2)^-$ respectively. the dark lines in both figures are respectively the graphs of $f_1(x_1)^- = \max\{w_{i1}d(v_i, x_1)^-, i = 1, 2, 3\}$ and $f_2(x_2)^- = \max\{w_{i2}d(v_i, x_2)^-, i = 1, 2, 3\}$. Problem P_{L2}' produces a lower bound 3.

4.4.4. A Lower Bounding Problem for the Multicenter Problem

Here, we develop a better lower bounding problem for a multicenter subproblem

P': $\underset{X \in S}{\text{Minimize}}\ f(X) = \max\{\ \ldots, w_{ij}d(v_i, x_j), \ldots, v_{jk}d(x_j, x_k), \ldots\}$.

Let $f_j(x_j) = \max\{v_{ij}d(v_i, x_j), i = 1, \ldots, m\}$, so that $f(X) = \max\{\ \max\{f_j(x_j), j = 1, \ldots, n\}, v_{jk}d(x_j, x_k), j < k\}$. Over $L_{[j]}$ – a segment in some edge, $f_j(x_j)$ is piecewise linear. The best PLC underestimate for each $f_j(x_j)$ over $L_{[j]}$ is its PLC supporting plane. Together with the $d(x_j, x_k)^-$ developed in this Subsection 4.4.1B, we obtain a PLC lower bounding problem for P', which is the best for P' discussed in this chapter.

Example 4.13. Consider again the instance of the multicenter problem in Example 4.12. The graphs of $f_1(x_1)$ and $f_2(x_2)$ are shown in Figure 4.17a and 4.17c respectively. The best PLC

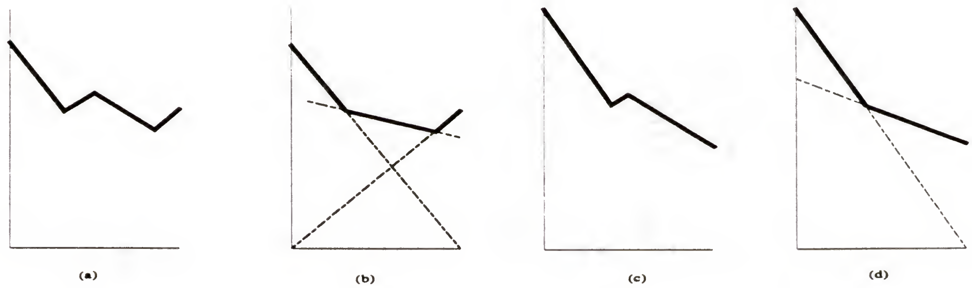


Figure 4.17 Graphs of f_j and p_j

underestimate of $f_1(x_1)$ is $p_1(x_1) = \max\{6(1-t(x_1)), 4t(x_1), (1/11)(-12t(x_1) + 48)\}$, where $(1/11)(-12t(x_1) + 48)$ is the line passing through the local minima of f_1. The best PLC underestimate for $f_2(x_j)$ is $p_2(x_2) = \max\{7(1-t(x_2)), -2t(x_2)+5\}$, where $-2t(x_2)+5$ is the line passing through local minima of f_2. The graphs of $p_1(\ )$ and $p_2(\ )$ are shown in Figure 4.17b and 4.17d, respectively. The $d(x_1, x_2)^-$ is the same as in Example 4.12. Hence, we can formulate a lower bounding problem: Minimize $\max\{6-6t_1, 4t_1, 48/11-(12/11)t_1, 7-7t_2, 5-2t_2, 3.5-3.5t_1, 3.5t_2 | 0 \le t_1, t_2 \le 1\}$, which produces a lower bound 3.43 (the optimal value of P' is 3.43).

Property 4.5. Let $p_j(x_j)$ be the best PLC underestimate of $f_j(x_j)$ over $L_{[j]}$. Then the lower bounding problem P_{L3}': Minimize $f(X)^- = \max\{\ \max\ \{p_1(x_1), \ldots, p_n(x_n)\}, \ldots\ v_{jk}d(x_j, x_k)^-, \ldots\}$ is the best for P' among the three types of lower bounding problems for P' considered so far.

Since each $f_j(x_j)$ is a piecewise linear function, the PLC supporting plane for $f_j(x_j)$ over $L_{[j]}$ is the PLC supporting plane of a set of points $(\mu_0, f_j(\mu_0)), \ldots, (\mu_p, f_j(\mu_p))$, where μ_0 and μ_p are the two end-points of $L_{[j]}$ and μ_i, $i = 2, \ldots, p-1$, are points in $B_j - A_j$ where $B_j = \{\beta \in L_{[j]} | w_{ij}d(v_i, \beta)$

$= w_{hj}d(v_h, \beta)$ for some h and i} is the set of bottleneck-points in $L_{[j]}$, and A_j ($A_j \subset B_j$) is the set of local maxima of $f_j(x_j)$ in $L_{[j]}$. One can find B_j and A_j in low order polynomial-time. Thus, the problem of finding the PLC supporting plane is a special case of finding the PLC y-dimension supporting plane for a set of points in E^2. In Appendix B.3, we give an $O(p^3)$ algorithm for the latter problem. Finally, note that this approach of using the best PLC underestimate for each individual $f_j(x_j)$ also applies to the multimedian problem where each $f_j(x_j) = \Sigma_i w_{ij}d(v_i, x_j)$. We do not discuss this extension here, since for the case of the multimedian problem, each of the best PLC underestimates $p_j(x_j)$ is the same as $\Sigma_i w_{ij}d(v_i, x_j)^-$, so that this "best PLC" approach will construct the same lower bounding problem P_{L2}' defined in Subsection 4.2.2.

4.5 Multifacility Problems on Grid Networks

In the previous section, lower bounding problems were useful only for subproblems at deep depths in the branching tree; that is, when a large percentage of location variables are restricted to edges. This is necessary for general cyclic networks, since the PLC underestimates of distance functions are only useful on subnetworks at the edge level. With a grid network, they are useful on much larger subnetworks, so that lower bounding problems are useful at much lower levels of the branching tree. In fact, as we have seen in Chapter 3, the rectilinear underestimate of d(. , .) exists on the entire original network. In this section, we will develop some additional PLC underestimates for d(. , .), which, in contrast to the rectilinear distance underestimate, progressively improve their approximation quality. We will also give lower bounding problems for various multifacility location problems defined on grid networks.

As in Chapter 3, let N_g denote the grid network, u_j denote a location variable on N_g with coordinates $(u_x, u_y) \in E^2$, and $r(p_1, p_2)$ $(= r_x(p_{x1}, p_{x2}) + r_y(p_{y1}, p_{y2}))$ denote the rectilinear distance between the two points. Since N_g is an embedding in E^2, N_g^n is an embedding in $E^n \times E^n$. A solution U in N_g^n is a vector in $E^n \times E^n$. Let U_x (U_y) denote the vector of the x-coordinates (y-coordinates) of U. Network N_g encloses a rectangle $[v_{xmin}, v_{xmax}] \times [v_{ymin}, v_{ymax}]$ in E^2.

For the rest of this chapter, we study problem P: Minimize$\{f(U) = c(D(U)) | U \in N_g^n\}$ and its subproblem P': Minimize$\{f(U) = c(D(U)) | U \in S \cap N_g^n\}$, with S a polytope in $E^n \times E^n$, defined later in this section.

4.5.1 Representing d(. , .) as Functions on E^2

In this subsection, we will see the following. Similar to rectilinear distances, the grid network distance d(. , .) can be represented as the sum of two functions $d_x(. , .)$ and $d_y(. , .)$ defined on N_g. Respectively the distance traveled along x-axis and y-axis, $d_x(. , .)$ and $d_y(. , .)$ can be explicitely represented as functions on E^2. But, $d_x(. , .)$ is not independent of the y-coordinates and neither is $d_y(. , .)$ independent of the x-coordinates. Another difficulty, which complicates our exposition and algorithms considerably, is that the analytical form of $d_x(. , .)$ and $d_y(. , .)$ are not unique. However, we still can use this "semi-separability" to develop some PLC underestimates for both $d_x(. , .)$ and $d_y(. , .)$.

Let $vl_1, ..., vl_p$, $vl_{i-1} < vl_i$, be the x-coordinates of vertical grid lines of N_g and $hl_1, ..., hl_q$, $hl_{i-1} < hl_i$, be the y-coordinates of horizontal grid lines of N_g. A vertex in the interior of some grid edge (i.e. not an intersection point) is a v-int vertex, if it is on a vertical grid line, or is a h-int vertex, if it is in a horizontal line. For any vertex v_i, let $vl_{[i]}$ and $vl_{[i]}'$ be the x-coordinates of the vertical grid lines adjacent to v_i, with $vl_{[i]} \leq vl_{[i]}'$ and $vl_{[i]} = vl_{[i]}'$ if v_i is on a vertical grid line. Let $hl_{[i]}$ and $hl_{[i]}'$ be similarly defined for the horizontal adjacent grid lines.

One concept that we will repeatedly encounter is the following. We say two points on N_g are semi-antipodal to each other if they are either (a) on two different vertical grid lines and both are in the interior of the same grid row; or (b) on two different horizontal grid lines and both are in the interior of the same grid column. Traveling between two semi-antipodal points on the grid network is like traveling from a point on one side of a rectangular obstacle to another point on the opposite side. The shortest distance between two semi-antipodal points thus is more than the rectilinear distance between them.

4.5.1.1. Type-I Distances

A type-I distance $d(v_i, u)$ can be represented as a function on N_g as follows. First of all,

$$d(v_i, u) = |v_{xi} - u_x| + |v_{yi} - u_y|, \quad \text{for any } u \in N_g, \text{ and any intersection vertex } v_i. \tag{4.9}$$

Furthermore,

$$d(v_i, u) = d_x(v_{xi}, u_x) + |v_{yi} - u_y|, \quad \text{for any } u \in N_g \text{ and any h-int vertex } v_i, \tag{4.10}$$

where $d_x(v_{xi}, u_x)$ is a real-valued function in E^2 defined as follows:

$$d_x(v_{xi}, u_x) = \begin{cases} |v_{xi} - u_x| & \text{if } u_x \le vl_{[i]}, \text{ or } u_x \ge vl_{[i]}', \text{ or } u_y = v_{yi} \\ \min\{v_{xi}+u_x-2vl_{[i]}, 2vl_{[i]}'-v_{xi}-u_x\} & \text{o/w (i.e. } vl_{[i]} < u_x < vl_{[i]}' \text{ and } u_y \ne v_{yi}), \end{cases} \tag{4.11}$$

Conditions in (4.11) tell when a point u is semi-antipodal to a h-int vertex v_i. If u is semi-antipodal to v_i, then traveling from u to v_i must first reach one of the vertical grid lines adjacent to v_i and u, and then from that grid line to v_i. Thus, the shortest distance traveled along the x-axis is the smaller of $u_x - vl_{[i]} + v_{xi} - vl_{[i]}$ and $vl_{[i]}' - u_x + vl_{[i]}' - v_{xi}$, or equivalently, is $\min\{v_{xi} + u_x - 2vl_{[i]}, 2vl_{[i]}' - v_{xi} - u_x\}$. On the other hand, since v_i is a h-int vertex, the distance traveled along the y-axis is $|v_{yi} - u_y|$ for any $u \in N_g$. Symmetric to (4.10) and (4.11), we have

$$d(v_i, u) = |v_{xi} - u_x| + d_y(v_{yi}, u_y), \quad \forall u \in N_g, \text{ and any v-int vertex } v_i, \tag{4.12}$$

where

$$d_y(v_{xi}, u_y) = \begin{cases} |v_{yi} - u_y| & \text{if } u_y \le hl_{[i]}, \text{ or } u_y \ge hl_{[i]}', \text{ or } u_x = v_{xi} \\ \min\{v_{yi}+u_y-2hl_{[i]}, 2hl_{[i]}'-v_{yi}-u_y\} & \text{o/w } (hl_{[i]} < u_y < hl_{[i]}' \text{ and } u_x \ne v_{xi}). \end{cases} \tag{4.13}$$

To express $d(v_i, u)$ in a more unified way, we use a simpler function in E^2 to capture all the cases. Define real-valued functions in E^1

$$\pi(z|a_1, a_2, a_3) = \min\{a_1 + z - 2a_2, 2a_3 - a_1 - z\} \text{ and}$$

$$\phi(z|a_1, a_2, a_3) = \max\{|a_1 - z|, \pi(z|a_1, a_2, a_3)\},$$

where a_1, a_2, and a_3 are some given real numbers. Figure 4.18a and 4.18b depict, respectively, the graphs of $|z - a|$ and $\phi(z|a_1, a_2, a_3)$, $a_2 < a_1 < a_3$.

Furthermore, for any given vertex v_i, define real-valued functions on E^2

$$\delta_x(v_{xi}, u_x) = \begin{cases} |v_{xi} - u_x| & \text{if } v_{yj} = u_y = hl_i \text{ for some i} \\ \phi(u_x\,|v_{xi}, vl_{[i]}, vl_{[i]}') & \text{o/w}, \end{cases}$$

$$\delta_y(v_{yi}, u_y) = \begin{cases} |v_{yi} - u_y| & \text{if } v_{xi} = u_x = vl_i \text{ for some i} \\ \phi(u_y \,|v_{yi}, hl_{[i]}, hl_{[i]}') & \text{o/w.} \end{cases}$$

Observation 4.6. $d(v_i, u) = \delta_x(v_{xi}, u_x) + \delta_y(v_{yi}, u_y)$ for any $u \in N_g$ and any vertex v_i.

Proof. To verify, compare $\delta_x(v_{xi}, u_x) + \delta_y(v_{yi}, u_y)$ case by case with $d(v, u)$ for the cases in (4.9), ..., (4.13). One property used in the comparison is that $\phi(u_x \,|v_{xi}, vl_{[i]}, vl_{[i]}') = \pi(u_x \,|v_{xi}, vl_{[i]}, vl_{[i]}') > |v_{xi} - u_x|$ if and only if $vl_{[i]} < u_x < vl_{[i]}'$. That is, $\phi(u_x \mid v_{xi}, vl_{[i]}, vl_{[i]}')$ is greater than $|v_{xi} - u_x|$ only when v_i is a h-int vertex and u_x is in the interior of the same grid column containing v_i. A similar property for $\phi(u_y \,|v_{yi}, hl_{[i]}, hl_{[i]}')$ is also used.

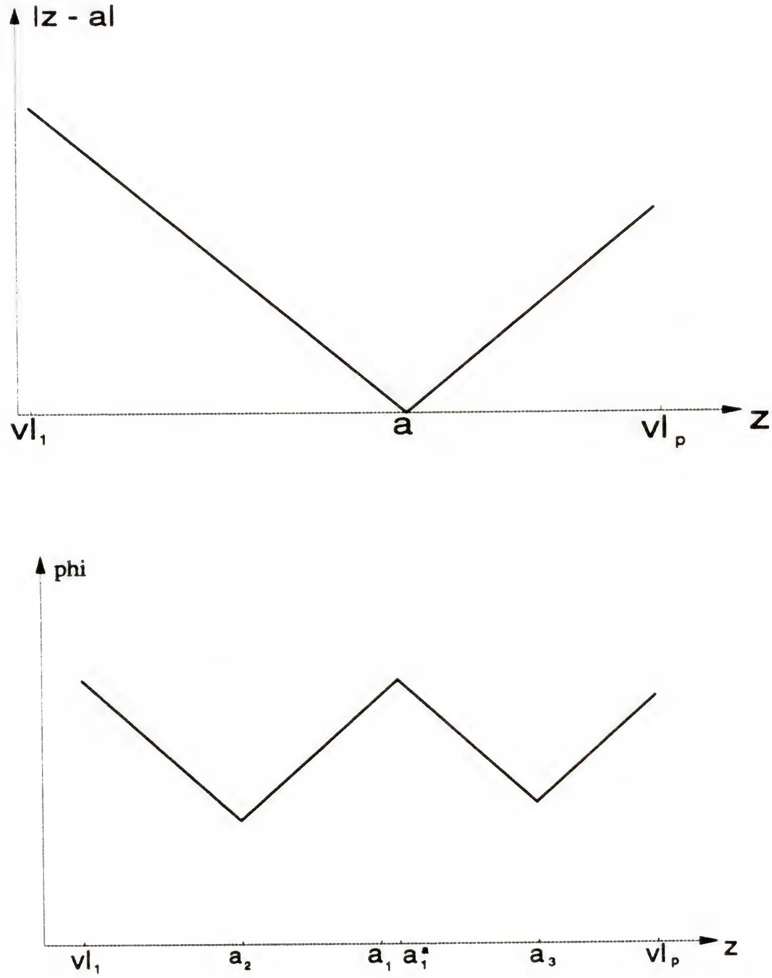


Figure 4.18 The Graphs of Functions ϕ and $|z - a|$

4.5.1.2. Type-II Distances

Similar to the type-I distance case, for any $(u_j, u_k) \in N_g^2$, if u_j and u_k are semi-antipodal inside grid column i, then

$$d(u_j, u_k) = \min\{u_{xj} + u_{xk} - 2vl_i, 2vl_{i+1} - u_{xj} - u_{xk}\} + |u_{yj} - u_{yk}|, \tag{4.14}$$

if u_j and u_k are semi-antipodal inside grid row i, then

$$d(u_j, u_k) = |u_{xj} - u_{xk}| + \min\{u_{yj} + u_{yk} - 2hl_i, 2hl_{i+1} - u_{yj} - u_{yk}\}, \tag{4.15}$$

otherwise

$$d(u_j, u_k) = |u_{xj} - u_{xk}| + |u_{yj} - u_{yk}|. \tag{4.16}$$

To express $d(u_j, u_k)$ with a simpler function,

First, define real-valued functions on E^2 as follows

$$\varphi_x(z_1, z_2) = \max\{|z_1 - z_2|, \tau_{x1}(z_1, z_2), ..., \tau_{x,p-1}(z_1, z_2)\},$$
$$\varphi_y(z_1, z_2) = \max\{|z_1 - z_2|, \tau_{y1}(z_1, z_2), ..., \tau_{y,q-1}(z_1, z_2)\},$$

where

$$\tau_{xi}(z_1, z_2) = \min\{z_1 + z_2 - 2vl_i, 2vl_{i+1} - z_1 - z_2\}, i = 1, \ldots, p-1, \text{ and}$$
$$\tau_{yi}(z_1, z_2) = \min\{z_1 + z_2 - 2hl_i, 2hl_{i+1} - z_1 - z_2\}, i = 1, \ldots, q-1.$$

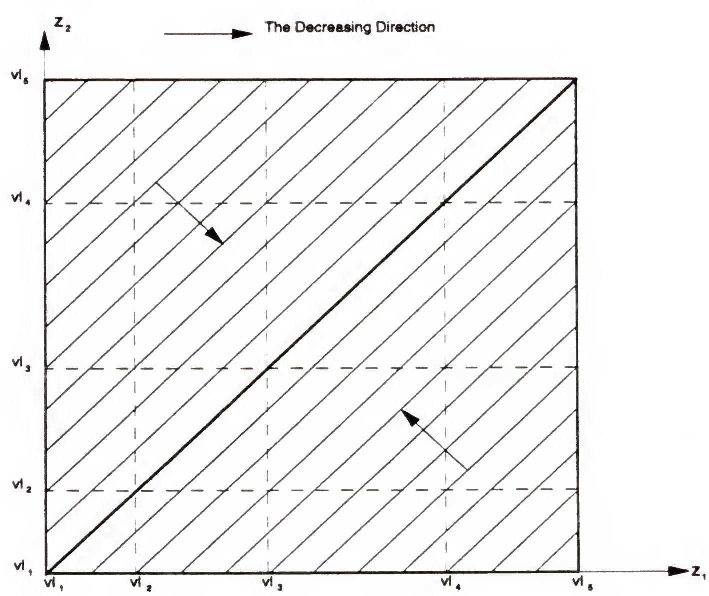


Figure 4.19 The Contour Set of $|z_1 - z_2|$

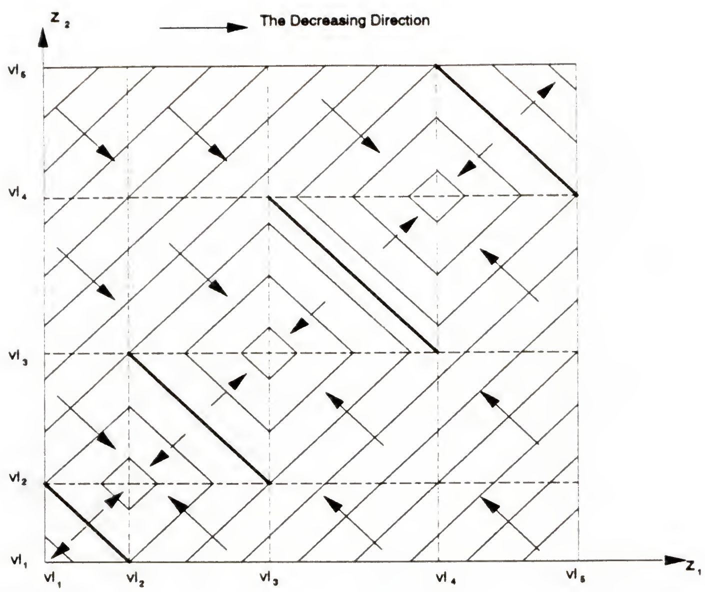


Figure 4.20 The Contour Set of φ

Figure 4.19 and 4.20 depict, respectively, the contours of $|z_1 - z_2|$ and $\varphi_x(z_1, z_2)$ over $[vl_1, vl_p]^2$ with p = 5. Since there is no structural difference between φ_x and φ_y, the contours of φ_y over $[hl_1, hl_q]^2$ are similar. Every $\tau_{xi}(z_1, z_2)$ has the property that $\tau_{xi}(z_1, z_2) > |z_1 - z_2|$ if and only if $(z_1, z_2) \in SR_i = (vl_i, vl_{i+1})\times(vl_i, vl_{i+1})$, so that over each open set SR_i, $\varphi_x(z_1, z_2) = \tau_{xi}(z_1, z_2)$, piecewise linear and concave; Over the rest of E^2, $\varphi_x(z_1, z_2) = |z_1 - z_2|$, linearly convex. Furthermore, each $\tau_{xi}(u_{xj}, u_{xk})$ is the distance traveled from u_j to u_k along the x-axis when u_j and u_k are semi-antipodal inside grid column i. Thus, $\varphi_x(u_{xj}, u_{xk})$ is the distance from u_j to u_k along the x-axis for all the cases of (u_j, u_k) on $N_g^2\times N_g^2$ except when u_j and u_k are on the same horizontal grid line and, at the same time, are in the interior of some grid column. The function φ_y has parallel properties. To summarize, define real-valued functions on $E^2\times E^2$ as follows:

$$\rho_x(u_{x1}, u_{x2}) = \begin{cases} |u_{x1} - u_{x2}| & \text{if } u_{y1} = u_{y2} = hl_i \text{ for some i} \\ \varphi_x(u_{x1}, u_{x2}) & \text{o/w} \end{cases}$$

and

$$\rho_y(u_{x1}, u_{x2}) = \begin{cases} |u_{y1} - u_{y2}| & \text{if } u_{x1} = u_{x2} = vl_i \text{ for some i} \\ \varphi_y(u_{y1}, u_{y2}) & \text{o/w.} \end{cases}$$

<u>Observation 4.7</u>. $d(u_j, u_k) = \rho_x(u_{xj}, u_{xk}) + \rho_y(u_{yj}, u_{yk})$ for any $(u_j, u_k) \in N_g^2$.

Proof. The proof is a straightforward case by case comparison between $\rho_x(u_{xj}, u_{xk}) + \rho_y(u_{yj}, u_{yk})$ and $d(u_j, u_k)$ for those cases listed in (4.14) to (E4.16). ∎

4.5.2 Definition of an L-Set S

Now, we define the type of solution subsets in the B&B scheme. For the general cyclic network case, we define L-sets directly on G^n. Here, we define a solution subset by defining a polytope S in $E^n \times E^n$. We call S an L-set, where L stands for linear. Since $N_g{}^n$ is an embedding in $E^n \times E^n$, $S \cap N_g{}^n$ is well-defined after defining S. In fact, S is defined based on some geometric terms involving N_g. We define S by defining $S_x \subset E^n$ and $S_y \subset E^n$, and letting $S = S_x \times S_y$. We assume that S_x and S_y have no structural differences, so that we only discuss S_x.

4.5.2.1. The Topology of δ_x and ρ_x

Now, we provide motivation for the way we define S_x. Similar to the approach for a general cyclic network, S_x will be defined by some hyperplanes, each of which is associated with some break points at which δ_x or ρ_x reaches it (local) maximum. Partitioning a solution subset with such a hyperplane reduces the number of local maxima in the resulting subsets.

Definition 4.4. (The LM (Local Maximum) Points in a Grid Line)

For a h-int vertex v_i, let $v_{xi}{}^a = vl_{[i]} + vl_{[i]}' - v_{xi}$. A point $s_i{}^a \in N_g$ is an LM point of v_i if $s_{xi}{}^a = v_{xi}{}^a$, and $s_i{}^a$ is not in the same horizontal grid line that contains v_i.

An LM point $s_i{}^a$ of a h-int vertex v_i is the antipodal point of v_i in the sense that there exists an $\varepsilon > 0$ such that $\phi_x(s_{xi}{}^a \mid v_{xi}, vl_{[i]}, vl_{[i]}') > \phi_x(p \mid v_{xi}, vl_{[i]}, vl_{[i]}')$ for any p in $[s_{xi}{}^a - \varepsilon, s_{xi}{}^a + \varepsilon]$, $p \neq s_{xi}{}^a$. A $s_{xi}{}^a$ is equivalent to point $a_1{}^a$ in Figure 4.18b. For example, if v_i is the middle point of the bottom horizontal grid line in a single cycle grid network. Then, v_i has an LM point in the middle of the top horizontal grid line. In general, for a h-int vertex v_i, each horizontal grid line that does not contain v_i contains exactly one of its LM points, so that v_i has exactly q−1 LM points. Due to the following remark, we include each $\{(u_x, u_y) \in E^1 \mid u_x = v_{xi}{}^a\}$ into the candidate hyperplanes.

Remark 4.4. For an h-int vertex v_i, $\delta_x(v_{xi}, u_x)$ is PLC over $[v_{xmin}, v_{xi}{}^a]$ and $[v_{xi}{}^a, v_{xmax}]$ for any given u_y.

Proof. From the definition, $\delta_x(v_{xi}, u_x) = |v_{xi} - u_x|$ if $u_y = v_{yi}$, and $\delta_x(v_{xi}, u_x) = \phi(u_x \,|v_{xi}, vl_{[i]}, vl_{[i]}')$ otherwise. For the first case, $\delta_x(v_{xi}, u_x)$ has a v-shaped graph and either $v_{xi}{}^a = v_{xmin}$ or $v_{xi}{}^a = v_{xmax}$. Thus, the remark is true. For the second case, $\delta_x(v_{xi}, u_x)$ has a double v-shaped graph similar to the one shown in Figure 18b. The $v_{xi}{}^a$ is the local maximum such that the graphs of $\delta_x(v_{xi}, u_x)$ over $[v_{xmin}, v_{xi}{}^a]$ and $[v_{xi}{}^a, v_{xmax}]$ are both v-shaped. The remark is true. ∎

Now, we study ρ_x. Define hyperplanes $H_i = \{(u_{x1}, u_{x2}) \in E^2 \mid u_{x1} + u_{x2} = vl_i + vl_{i+1}\}$, i = 1, …, p−1, where p is the number of vertical grid lines in N_g. As one can see from Figure 4.20, H_i is the hyperplane coinciding with the line segment L_{Hi} that has end points (vl_i, vl_{i+1}) and (vl_{i+1}, vl_i). Also from Figure 4.20, $\varphi_x(u_{x1}, u_{x2})$ reaches its (local) maximum at points on each L_{Hi}. Since $\rho_x(u_{xj}, u_{xk}) = \varphi_x(u_{xj}, u_{xk})$ for many cases, $\rho_x(u_{xj}, u_{xk})$ often reaches its local maximum at the points on these L_{Hi}'s. Thus, we include the H_i's as candidate hyperplanes. We also use $\{u \in E^2 \mid u_x = vl_l\}$, l = 1, … p, to confine location variables to vertical grid lines and use $\{u \in E^2 \mid u_x = v_{xmin}\}$ and $\{u \in E^2 \mid u_x = v_{xmax}\}$ to confine location variables within $[v_{xmin}, v_{xmax}]$.

4.5.2.2. Defining S_x

Let $H_i^+ = \{(u_{x1}, u_{x2}) \in E^2 \mid u_{x1} + u_{x2} \geq vl_i + vl_{i+1}\}$, i = 1, …, p−1,

$H_i^- = \{(u_{x1}, u_{x2}) \in E^2 \mid u_{x1} + u_{x2} \leq vl_i + vl_{i+1}\}$, i = 1, …, p−1,

$H = \{H_1, \ldots, H_{p-1}\}$, $H^- = \{H_1^-, \ldots, H_{p-1}^-\}$, and $H^+ = \{H_1^+, \ldots, H_{p-1}^+\}$,

$P_S = \{v_{xi}{}^a \mid v_i \text{ is an h-int vertex}\} \cup \{vl_i \mid i = 1, \ldots p\}$.

Definition 4.5. An S_x is defined with some a_j and b_j in P_S, $B \subseteq \{(j, k) \mid 1 \leq j < k \leq n\}$, and K_{jk} an intersection of some half-planes in $H^+ \cup H^-$ for each (j, k) in B, such that

$$S = \{U_x \in E^n \mid a_j \leq u_j \leq b_j, \text{ for every j, and } (u_{xj}, u_{xk}) \in K_{jk} \text{ for } (j, k) \in B\}.$$

Example 4.14. Let P be a 3-facility instance defined on the grid network N_g shown in Figure 4.21, with width and length three units. The grid columns and grid rows are equally spaced. Let v_1 be the vertex in the middle of the second horizontal grid line. According to the definition of LM points (Definition 4.4), $v_{x1}{}^a = 1.5$. The following point sets satisfy the definition of S_x:

a. $S_x = \{(u_{x1}, u_{x2}, u_{x3}) \mid 0 \leq u_{x1} \leq 1.5, 0 \leq u_{x2} \leq 3, 0 \leq u_{x3} \leq 3\}$

b. $S_x = \{(u_{x1}, u_{x2}, u_{x3}) \mid 1.5 \leq u_{x1} \leq 3, 0 \leq u_{x2} \leq 3, 0 \leq u_{x3} \leq 3\}$

c. $S_x = \{(u_{x1}, u_{x2}, u_{x3}) \mid 1 \leq u_{x1} \leq 2, 1 \leq u_{x2} \leq 2, 0 \leq u_{x3} \leq 3, u_{x1} + u_{x2} \leq 3 \ (= vl_2 + vl_3)\ \}$

d. $S_x = \{(u_{x1}, u_{x2}, u_{x3}) \mid 1.5 \leq u_{x1} \leq 3, 1 \leq u_{x2} \leq 3, 0 \leq u_{x3} \leq 3, 3 \leq u_{x1} + u_{x2} \leq 5\}$.

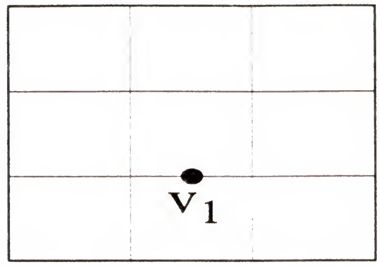


Figure 4.21 An Example Grid Network

Not every half-plane involved in an S_x is necessarily binding. Similar to the L-set defined in the last section, the binding constraints of an S_x must satisfy the following

Constraint-Description 4.2

(a) for each j, there are at most two single-variable half-planes involving variable u_{xj};

(b) for each (j, k), there are at most 2 two-variable half-planes involving both variables u_{xj} and u_{xk}; each such half-plane is in $H^{+} \cup H^{-}$.

In the constraint types and their properties, an L-set defined here is very similar to the L-set defined in Section 4.4. To identify all the binding constraints, we can design an algorithm similar to the one given in Appendix B.1. Thus, from now on, we assume that all the binding constraints for S_x are known. In particular, we assume that $lb_j = \min\{u_{xj} | U_x \in S_x\}$, $rb_j = \max\{u_{xj} | U_x \in S_x\}$, and $X_{jk} = \{(u_{xj}, u_{xk}) \mid U_x \in S_x\}$ has explicite form.

Observation 4.8. An X_{jk} is either a triangle, a quadrilateral, a pentagon, or a hexagon in E^2.

Proof. First of all, X_{jk} is inside rectangle $[lb_j, rb_j] \times [lb_k, rb_k]$. The binding two-variable half-planes, if there are any, will reduce X_{jk} to one of the geometric regions listed above. ∎

After removing all the redundant constraints, we can express S_x as

$$S_x = \{U_x \in E^n \mid lb_j \le u_{xj} \le rb_j, \text{ for each j, and } (u_{xj}, u_{xk}) \in X_{jk} \text{ for all } j < k\}.$$

Example 4.15. In Example 4.14, the S_x in c is a triangle; the S_x in d is a hexagon.

4.5.3. Representation Uniqueness and/or PLC

In Observation 4.6 and 4.7 we established, respectively, that $d(v_i, u) = \delta_x(v_{xi}, u_x)+\delta_y(v_{yi}, u_y)$ for any $u \in N_g$ and $d(u_j, u_k) = \rho_x(u_{xj}, u_{xk}) + \rho_y(u_{yj}, u_{yk})$ for any $(u_j, u_k) \in N_g^2$. In Subsection 4.5.2, we see that it is possible to define solution subsets based on the topology of functions δ_x,

δ_y, ρ_x, and ρ_y, when the corresponding functions have unique forms over the entire domains. It is equally important to know whether a distance function has a unique form over the entire domain when we consider its PLC underestimates. Thus, in this subsection, we give some sufficient conditions for an L-set S for determining (a) whether a function has a unique functional representation over $S \cap N_g^n$; (b) whether a function is PLC over $S \cap N_g^n$. Again, we only need to discuss conditions for functions δ_x and ρ_x.

Throughout the remainder of this section, let

$lb_j = \min\{u_{xj} \mid U_x \in S_x\}$, $rb_j = \max\{u_{yj} \mid U_x \in S_x\}$, and $X_{jk} = \{(u_{xj}, u_{xk}) \mid U_x \in S_x\}$

$bb_j = \min\{u_{yj} \mid U_y \in S_y\}$, $tb_j = \max\{u_{yj} \mid U_y \in S_y\}$, and $Y_{jk} = \{(u_{xj}, u_{xk}) \mid U_y \in S_y\}$.

First, we give some sufficient conditions for $\delta_x(v_{xi}, u_{xj})$ to have a unique form for all $U \in S \cap N_g^n$. The conditions are not necessarily mutually exclusive.

Observation 4.9. Let v_i be a vertex on N_g and j be a location variable index.

(a) If $v_{yj} \notin [bb_j, tb_j]$, then $\delta_x(v_{xi}, u_{xj}) = \phi(u_{xj} \mid v_{xi}, vl_{[i]}, vl_{[i]}')$ for any $U \in S \cap N_g^n$;

(b) $\delta_x(v_{xi}, u_{xj}) = |v_{xi} - u_{xj}|$ for any $U \in S$, if at least one of the following conditions is true:

(b.1) $v_{xi} = vl_l$ for some vertical grid line l,

(b.2) either $rb_j \le vl_{[i]}$ or $vl_{[i]}' \le lb_j$,

(b.3) $v_{yj} = bb_j = tb_j$.

Proof. For Case (a), since $u_{yj} \neq v_{yi}$ for any $U \in S \cap N_g^n$, thus, from the definition, $\delta_x(v_{xi}, u_{xj}) = \phi(u_{xj} \mid v_{xi}, vl_{[i]}, vl_{[i]}')$. Condition (b.1) says that v_i is either an intersection vertex or a v-int vertex. Condition (b.2) implies that u_j cannot be semi-antipodal to v_i for any U. Condition (b.3) says that in S, u_j is restricted to the same grid line that contains v_i. We see that under any one of these conditions, $\delta_x(v_{xi}, u_{xj}) = |v_{xi} - u_{xj}|$. ∎

To see how general these conditions are, note that a horizontal grid line which does not contain v_i satisfies condition (a); a region in E^2, which can be separated from v_i by a vertical grid line, satisfies condition (b.2). In the following, we give some sufficient conditions for $\rho_x(u_{xj}, u_{xk})$ to have a unique form over $S \cap N_g^n$.

Observation 4.10. Let S be an L-set. For a variable index pair (j, k),

(a) If Y_{jk} contains no point (hl_i, hl_i), $i = 1, \ldots q-1$, then $\rho_x(u_{xj}, u_{xk}) = \varphi_x(u_{xj}, u_{xk})$, $\forall U \in S \cap N_g^n$;

(b) $\rho_x(u_{xj}, u_{xk}) = |u_{xj} - u_{xk}|$ for any $U \in S \cap N_g^n$ if at least one of the following conditions is true:

(b.1) there exists a vl_i, $1 \le i \le p$, that separates $[lb_j, rb_j]$ from $[lb_k, rb_k]$

(i.e. either $rb_j \le vl_i \le lb_k$ or $rb_k \le vl_i \le lb_j$),

(b.2) $bb_j = tb_j = bb_k = tb_k$.

Proof. If u_j and u_k are in the same horizontal grid line, then $(u_{yj}, u_{jk}) = (hl_i, hl_i)$ for some i. Condition (a) guarantees that u_j and u_k are not in the same horizontal grid line for any $U \in S \cap N_g^n$. Therefore, from the definition of ρ_x, $\rho_x(u_{xj}, u_{xk}) = \varphi_x(u_{xj}, u_{xk})$.

Conditions (b.1) guarantees that there is no $U \in S \cap N_g^n$ with u_j and u_k semi-antipodal; condition (b.2) implies that both u_j and u_k are restricted to the same horizontal grid line. Therefore, under (b.1) and/or (b.2), $\rho_x(u_{xj}, u_{xk}) = |u_{xj} - u_{xk}|$.

Example 4.16. Consider a 2-facility instance defined on the grid network shown in Figure 4.21. Let S be an L-set with $S_x = \{(u_{x1}, u_{x2}) \in E^2 | 0 \le u_{x1} \le 1.5, 0 \le u_{x2} \le 3, \text{ and } u_{x1}+u_{x2} \ge vl_2+vl_3 (= 3)\}$. In this case, $X_{12} = \{(u_{x1}, u_{x2}) \in E^2 | 0 \le u_{x1} \le 1.5, 1.5 \le u_{x2} \le 3, \text{ and } u_{x1} + u_{x2} \ge vl_2+vl_3 (= 3)\}$. On the other hand, $(vl_i, vl_i) = (i-1, i-1)$, $i = 1, 2, 3$, and 4. Set X_{12} contains no (vl_i, vl_i).

Once δ_x (or ρ_x) is known to have a unique form, determining whether it is PLC is straight - forward. Thus, Observations 4.9 and 4.10 are sufficient conditions for when a δ_x (or ρ_x) is PLC over $S \cap N_g^n$.

4.5.4 The PLC Underestimates of Function d(. , .) on $S \cap N_g^n$

Respectively in Observation 4.6 and 4.7, $d(v_i, u_j) = \delta_x(v_{xi}, u_{xj}) + \delta_y(v_{yi}, u_{yj})$ $\forall U \in S \cap N_g^n$ and $d(u_j, u_k) = \rho_x(u_{xj}, u_{xk}) + \rho_y(u_{xj}, u_{xk})$, $\forall U \in S \cap N_g^n$. We thus construct a PLC underestimate of d(. , .) on $S \cap N_g^n$ by constructing PLC underestimates of δ_x, δ_y, ρ_x, and ρ_y over $S \cap N_g^n$. We only consider δ_x and ρ_x, since the methods apply to δ_y and ρ_y with only notation changes.

4.5.4.1. The PLC Underestimate of Type-I Distance

Now we discuss finding an underestimate for δ_x over S. The universal underestimate is the rectilinear distance. When Observation 4.9(a) does not hold, either $\delta_x(v_{xi}, u_{xj}) = |v_{xi} - u_{xj}|$, as in

Observation 4.9 (b.1) and (b.2), or $\delta_x(v_{xi}, u_{xj})$ has no unique expression. It is thus only possible to make improvement when Observation 4.9(a) is true. In this case, $\delta_x(v_{xi}, u_{xj}) = \phi(z \mid a_1, a_2, a_3)$ over an interval [a, b] (with $z = u_{xj}$, $a_1 = v_{xi}$, $a_2 = vl_{[i]}$, $a_3 = vl_{[i]}'$, $a = lb_j$, $b = rb_j$). We hence only show how to find a PLC underestimate for $\phi(z \mid a_1, a_2, a_3)$ over [a, b]. As Figure 4.18b indicates, when $a_2 < a_1 < a_3$, $\phi(z \mid a_1, a_2, a_3)$ is a piecewise linear function of a double v-shaped graph. The shape of its graph over [a, b] depends on a and b. Nevertheless, the best PLC underestimate is the best PLC supporting plane of $\phi(z \mid a_1, a_2, a_3)$ over [a, b]. Figure 4.22 illustrates all the cases when the best PLC supporting plane is a nontrivial improvement for ϕ over [a, b] (in comparison to $|z - a_1|$).

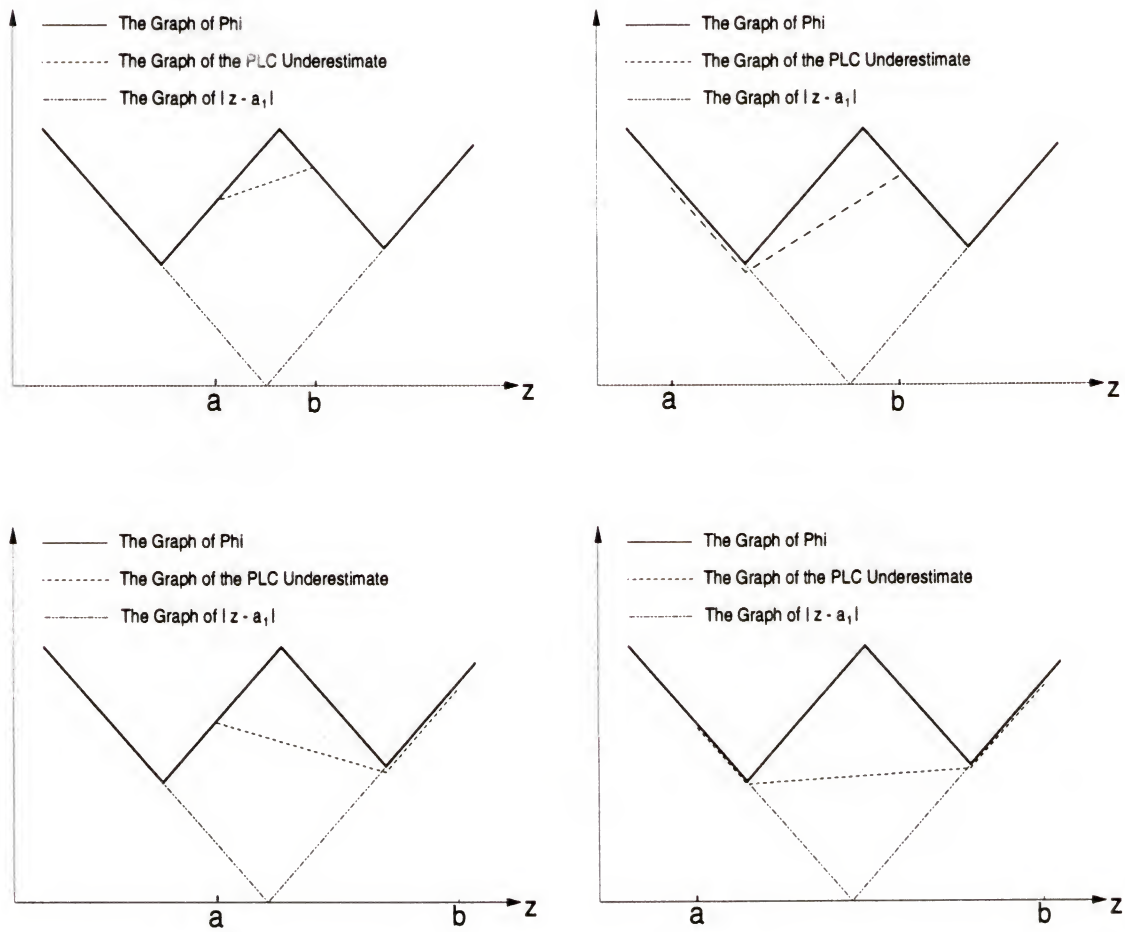


Figure 4.22 The PLC Underestimates

We now summarize the method as the following.

Procedure 4.2. (Constructing $\delta_x(v_{xi}, .)^-$)

If (a) of Observation 4.9 is true, then let $\delta_x(v_{xi}, .)^-$ be the best PLC supporting plane of the corresponding $\phi(. \,|v_{xi}, vl_{[i]}, vl_{[i]}')$ over $[lb_j, rb_j]$; Otherwise, let $\delta_x(v_{xi}, .)^- = r_x(v_{xi}, .)$;

4.5.4.2. The Underestimates of Type-II Distances

Again, the universal underestimate of $\rho_x(u_{xj}, u_{xk})$ is $|u_{xj} - u_{xk}|$. For improved underestimate of $\rho_x(u_{xj}, u_{xk})$ over X_{jk}, we only need consider Observation 4.10(a) when $\rho_x(u_{xj}, u_{xk}) = \varphi_x(u_{xj}, u_{xk})$ for all $U \in S\cap N_g^n$), since otherwise either $\rho_x(u_{xj}, u_{xk}) = |u_{xj} - u_{xk}|$ for all $U \in S\cap N_g^n$ (as in Observation 4.10 (b.1) or (b.2)), or $\rho_x(u_{xj}, u_{xk})$ does not have a unique form. Let $R_{jk}' = [lb_j, rb_j]\times[lb_k, rb_k]$. From Observation 4.8, $X_{jk} \subseteq R_{jk}'$. It is thus sufficient to give a method of finding a PLC underestimate of $\varphi_x(.\,,.)$ over an arbitrary rectangle $R \subseteq [vl_1, vl_p]^2$.

From Figure 4.20, φ_x is not convex only over those open square regions $SR_i = \{(z_1, z_2) \in E^2 \mid vl_i < z_1 < vl_{i+1}, vl_i < z_2 < vl_{i+1}\}$, i = 1, ..., p−1. Let CR_i be the closure of $R\cap SR_i$. Function $\varphi_x(.\,,.)$ over CR_i is piecewise linear and concave if and only if $SR_i\cap H_i \neq \emptyset$. The approach for an underestimate of φ_x over R is to first obtain, respectively, an underestimate for φ_x over each CR_i where φ_x is nonlinear and concave, and then combine these underestimates together with underestimate $|z_1 - z_2|$ to form a general one.

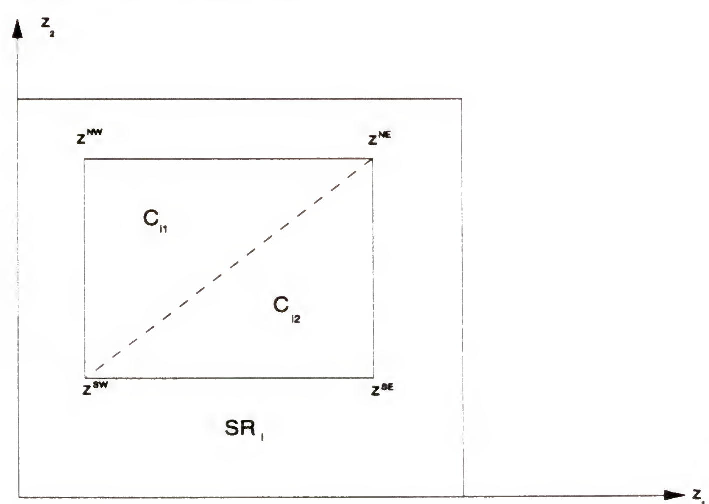


Figure 4.23 Corner Points and Convex Hulls

Now, we construct an underestimate of φ_x over a nonempty CR_i. Let z^{NE}, z^{NW}, z^{SE}, and z^{SW} be the four corner points of CR_i as shown in Figure 4.23. Let C_{i1} be the convex hull of z^{NE}, z^{NW}, and z^{SW}, and C_{i2} the convex hull of z^{NE}, z^{SE}, and z^{SW}. Let $l_{i1}(z_1, z_2)$ be the linear plane in E^3 defined by points $(z^{NE}, \varphi_x(z^{NE}))$, $(z^{NW}, \varphi_x(z^{NW})$, and $(z^{SW}, \varphi_x(z^{SW}))$, and let $l_{i2}(z_{x1}, z_{x2})$ be the linear plane defined by $(z^{NE}, \varphi_x(z^{NE}))$, $(u^{SE}, \varphi_x(z^{SE}))$, and $(z^{SW}, \varphi_x(z^{SW}))$. Finally, define the PLC function

$$pl_i(z_1, z_2) = \max\{l_{i1}(z_1, z_2), l_{i2}(z_1, z_2)\}.$$

Now we show that, over CR_i, pl_i is a nontrivially better underestimate of φ_x than $|z_1 - z_2|$.

<u>Lemma 4.3</u>. $pl_i(z_1, z_2) = l_{i1}(z_1, z_2)$, $\forall$ $(z_1, z_2) \in C_{i1}$ and $pl_i(z_1, z_2) = l_{i2}(z_1, z_2)$, $\forall$ $(z_1, z_2) \in C_{i2}$.

<u>Proof</u>. See Appendix B.4.<u>Theorem 4.1</u>. Let KR_i be the closure of SR_i. For any $(z_1, z_2) \in CR_i$,

a) $|z_1 - z_2| \le pl_i(z_1, z_2)$ with equality holding if and only if (z_1, z_2) is a boundary point of KR_i, or $CR_i = KR_i$;

b) $pl_i(z_1, z_2) \le \varphi_x(z_1, z_2)$.

<u>Proof</u>. From Lemma 4.3, $pl_i(z_1, z_2) = l_{i1}(z_1, z_2)$ for any $(z_1, z_2) \in C_{i1}$. The function l_{i1} over C_{i1} is a triangle in E^3 above the convex function $|z_1 - z_2|$ and below the concave function $\varphi_x(z_1, z_2)$. Thus, we have $|z_1 - z_2| \le pl_{i1}(z_1, z_2) \le \varphi_x(z_1, z_2)$, for any $(z_1, z_2) \in C_{i1}$. Similarly, $|z_1 - z_2| \le pl_i(z_1, z_2) \le \varphi_x(z_1, z_2)$, for any $(z_1, z_2) \in C_{i2}$. Since $C_{i1} \cup C_{i2} = CR_i$, we have the theorem. ■

<u>Lemma 4.4</u>. The underestimate function $pl_i(z_1, z_2)$ defined on CR_i is not greater than $|z_1 - z_2|$ outside of CR_i (i.e. over region $R - CR_i$).

<u>Proof</u>. See Appendix B.4.

Now, define

$$pl(z_1, z_2) \equiv \max\{|z_1 - z_2|, \max\{pl_i(z_1, z_2) \mid \text{for every } CR_i \neq \emptyset\}\}.$$

<u>Theorem 4.2</u>. $pl(z_1, z_2)$ is an underestimate for $\varphi_x(z_1, z_2)$ over R.

<u>Proof</u>. Let (z_1, z_2) be an arbitrary point in R. We have either $(z_1, z_2) \in CR_i$ for some i or $(z_1, z_2) \notin CR_i$ for all i. For the first case, from Theorem 4.1a, we have $|z_1 - z_2| \le pl_i(z_1, z_2)$. From Lemma 4.4, we have $pl_h(z_1, z_2) \le |z_1 - z_2|$ for any other h. Thus, $pl(z_1, z_2) = pl_i(z_1, z_2)$. From Theorem 4.1b, we have $pl_i(z_1, z_2) \le \varphi_x(z_1, z_2)$. For the second case, from Lemma 4.4, we have $pl_i(z_1, z_2) \le |z_1 - z_2|$ for any i. Thus, $pl(z_1, z_2) = |z_1 - z_2| \le \varphi_x(z_1, z_2)$. ■

Note also that $pl(.)$ will be φ_x itself if the latter is linear over R.

The following procedure summarizes the steps of constructing an improved underestimate for $\rho_x(u_{xj}, u_{xk})$ over X_{jk}.

Procedure 4.3. (Constructing $\rho_x(. , .)^-$)

If (a) of Observation 4.9 is true, then $\rho_x(. , .)^- = pl(. , .)$; otherwise, let $\rho_x(. , .)^- = r_x(. , .)$;

4.5.5. The Lower Bounding Problems

Now, we consider lower bounding problems for a subproblem

P': Minimize $\{f(U) = c(D(U)) \mid U \in S \cap N_g^n\}$.

With the PLC distance underestimates, an obvious lower bounding problem for P' is

P_L': Minimize $\{f(X)^- = c(\ldots, d(v_i, u_j)^-, \ldots, d(u_j, u_k)^- \ldots) \mid U \in S\}$, where $d(v_i, u_j)^- = \delta_x(v_{xi}, u_{xj})^- + \delta_y(v_{yi}, u_{yj})^-$ and $d(u_j, u_k)^- = \rho_x(u_{xj}, u_{xk})^- + \rho_y(u_{yj}, u_{yk})^-$. Problem P_L' can be transformed into a convex programing problem with linear constraints, since function c is convex. As in Section 4.4, we can further obtain a LP (Linear Program) lower bounding problem for P' with subgradients evaluated at some extreme points of $S \cap N_g^n$.

In the remainder of this section, we will focus on two multifacility problems – the multimedian problem and the multicenter problem, since their respective objective function structures enable us to develop better lower bounding problems.

4.5.6. The Multimedian Lower Bounding Problems

The multimedian problem is P: Minimize $\{f(U) \mid U \in N_g^n\}$, where $f(U) = \Sigma_j f_j(u_j) + f_{NN}(U)$ with $f_j(u_j) = \Sigma_i w_{ij} d(v_i, u_j)$ and $f_{NN}(U) = \Sigma_{j<k} v_{jk} d(u_j, u_k)$. A subproblem is P': Minimize $\{f(U) \mid U \in S \cap N_g^n\}$, where $S = S_x \times S_y$ with S_x and S_y some polytopes in E^n defined in Subsection 4.5.2. From Observations 4.6 and 4.7, we express P' as Minimize $\{f_x(U) + f_y(U) \mid U \in S \cap N_g^n\}$, where $f_x(U) = \Sigma_{i,j} w_{ij} \delta_x(v_{xi}, u_{xj}) + \Sigma_{j<k} v_{jk} \rho_x(u_{xj}, u_{xk})$ and $f_y(U) = \Sigma_{i,j} w_{ij} \delta_y(v_{yi}, u_{yj}) + \Sigma_{j<k} v_{jk} \rho_y(u_{yj}, u_{yk})$.

Construct a lower bounding problem P_L' for P' as the following. For the given S, let I_j be the set of vertex indices such that for each $i \in I_j$, v_i and u_j satisfy at least one of the conditions (for a unique form) in Observation 4.9. In other words, for each $i \in I_j$, $\delta_x(v_{xi}, u_{xj})$ over $S \cap N_g^n$ has exactly one of the forms given in the definition of δ_x. Let $I_j' = I - I_j$, where I is the set of all the vertex indices.

Observation 4.11. For each $i \in I_j$, $\delta_x(v_{xi}, u_{xj})$ is a function of u_{xj} only.

Proof. For any $i \in I_j$, either $\delta_x(v_{xi}, u_{xj}) = \phi(u_{xj}|v_{xi}, vl_{[i]}, vl_{[i]}')$ or $\delta_x(v_{xi}, u_{xj}) = |v_{xi} - u_{xj}|$ for all the $U \in S \cap N_g^n$. Thus, the conclusion is true. ∎

Now, let $f_{xj}(u_{xj})^- = \Sigma\{w_{ij}\delta_x(v_{xi}, u_{xj})) \mid i \in I_j\} + \Sigma\{w_{ij}|v_{xi} - u_{xj}| \mid i \in I_j'\}$ and let $f_{yj}(u_{yj})^-$ be smilarly defined. We have a lower bouding problem

$$P'': \underset{U_x \in S_x,\ U_y \in S_y}{\text{Minimize}}\ \Sigma_j f_{xj}(u_{xj})^- + \Sigma_{j<k}\rho_x(u_{xj}, u_{xk})^- + \Sigma_j f_{yj}(u_{yj})^- + \Sigma_{j<k}\rho_y(u_{yj}, u_{yk})^-.$$

It is clear that problem P_L' can be decomposed into two independent problems

$$P_x'': \underset{U_x \in S_x,}{\text{Minimize}}\ \Sigma_j f_{xj}(u_{xj})^- + \Sigma_{j<k}\rho_x(u_{xj}, u_{xk})^- \text{ and } P_y'': \underset{U_y \in S_y,}{\text{Minimize}}\ \Sigma_j f_{yj}(u_{yj})^- + \Sigma_{j<k}\rho_y(u_{yj}, u_{yk})^-.$$

We can find linear programming lower bounding problems for P_x'' and P_y''. Here, we only construct the former, as the latter is totally parallel.

Note that since $\delta_x(v_{xi}, u_{xj})$ and $|v_{xi} - u_{xj}|$ are piecewise linear, $f_{xj}(u_{xj})^-$ is piecewise linear over $[lb_j, rb_j]$. It can be shown that $f_{xj}(u_{xj})^-$ has at most m break-points in $[lb_j, rb_j]$. Thus, we can use the procedure given in Appendix B.3 to construct its PLC supporting plane over $[lb_j, rb_j]$. Let pl_{xj} denote this PLC supporting plane. We then have a linear program lower bounding problem

P_{xL}'': $\underset{U_x \in S_x}{\text{Minimize}}\ \Sigma_j\, pl_{xj}(u_{xj}) + \Sigma_{j<k}\rho_x(u_{xj}, u_{xk})^-$. Let P_{yL}'' denote the corresponding part for P_y''.

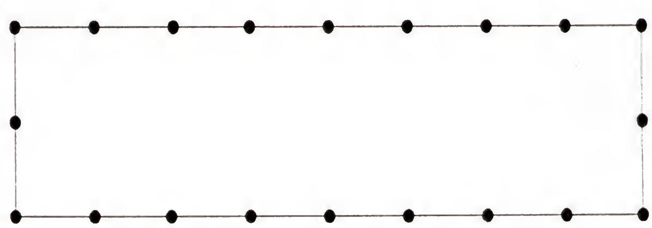


Figure 4.24 An Example Grid Network

Example 4.17. Consider a 3 new facility instance of P on the network G in Figure 4.24. The network has identical edge lengths of 1. The vertices are numbered from 1 to 20 counter - clockwise starting from the lower left corner. The weights are given below,

w_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	5	2	1	4	6	1	2	3	1	2	0	4	6	1	4	4	2	6	1	1	$v_{12} = 10$
2	2	2	2	7	2	5	6	1	4	4	1	2	0	9	2	4	2	2	7	8	$v_{13} = 10$
3	3	2	1	2	3	2	4	2	4	5	2	5	5	5	2	4	2	2	1	0	$v_{23} = 10$

A subproblem of P' has U restricted to $S = S_x \times S_y$ where $S_x = \{(u_{x1}, u_{x2}, u_{x3}) \mid 0 \le u_{x1} \le 4, 0 \le u_{x2} \le 8, 0 \le u_{x3} \le 8\}$ and $S_y = \{(u_{y1}, u_{y2}, u_{y3}) \mid u_{y1} = 0, u_{y2} = 2, 0 \le u_{y3} \le 2\}$. In words, new facility 1 is restricted to the left half of the bottom horizontal grid line of G, new facility 2 is restricted to the top horizontal grid line, and new facility 3 is unrestricted. For this subproblem, we have

$$I_1 = I_2 = I = \{1, \ldots, 20\}, \text{ and } I_3 = \varnothing,$$
$$pl_{x1}(u_{x1}) = 5.5u_{x1} + 209,\ pl_{x2}(u_{x2}) = 9.25u_{x2} + 215,$$
$$pl_{x3}(u_{x3}) = \max\{-48u_{x3} + 274, -40u_{x3} + 266, -34u_{x3} + 254, -22u_{x3} + 218, -12u_{x3} + 178, 2u_{x3} + 108, 20u_{x3}, 24u_{x3} - 28\}$$
$$\rho_x(u_{x1}, u_{x2})^- = \max\{u_{x1}, u_{x2} - u_{x1}\},\ \rho_x(u_{x1}, u_{x3})^- = |u_{x1} - u_{x3}|, \text{ and}$$
$$\rho_x(u_{x2}, u_{x3})^- = |u_{x2} - u_{x3}|.$$

Let $f_{xj}(u_{xj})'$ be the function obtained by replacing every function $\delta_x(v_{xi}, u_{xj})$ in $f_{xj}(u_{xj})$ with $\delta_x(v_{xi}, u_{xj})^-$; $h_{xj}(u_{xj})$ be the function obtained by replacing every $\delta_x(v_{xi}, u_{xj})$ with $|v_{xi} - u_{xj}|$. The graphs of $f_{xj}(\,)$, $f_{xj}(\,)'$, $pl_{xj}(\,)$, and $h_{xj}(\,)$ over $[lb_j, rb_j]$, j = 1, 2, are shown in Figures 4.25(a) and (b), as solid lines, dotted lines, dashed lines, and the dash-dot lines, respectively. Since u_3 is unrestricted, every $\delta_x(v_{xi}, u_{x3})^- = |v_{xi} - u_{x3}|$ and $I_3' = I - I_3 = I$. Thus, $f_{x3}(.) = pl_{x3}(.) = h_{x3}(.)$. Problem P_{xL}'' is a LP problem with 5 variables and 14 constraints. Lower bounding problem P_{yL}'' can be constructed similarly. Since, in S, u_{y1} and u_{y2} are fixed and u_{y3} is unrestricted, we have P_{yL}'': Minimize $\Sigma_i w_{i3}|v_{yi} - u_{y3}| + C$, where $C = \Sigma_i w_{i1} v_{yi} + \Sigma_i w_{i2}|v_{yi} - 2|$. The following Table 4.3 summarizes the minimal objective values of various problems (In the table, P_{rx}' and P_{ry}' are the rectilinear lower bounding problems for P_x' and P_y' respectively).

Table 4.3 The Objective Values

Problems	P'	P_{xL}''	P_{rx}'	P_{yL}''	P_{ry}'
Obj. Values	922	635	510	224	224

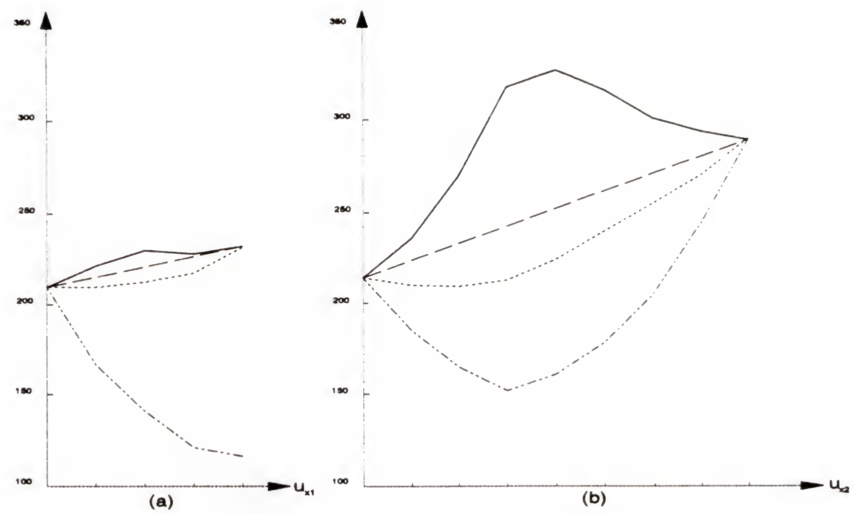


Figure 4.25 The Graphs of Various Functions

From Table 4.3, using lower bounding problems P_{xL}" and P_{yL}" produces a lower bound 859 which is significantly larger than the lower bound value 734 produced by lower bounding problem P_r – the rectilinear lower bounding problem.

As for the size of P_{xL}", each $pl_{xj}(u_{xj})$ is the maximum of at most m linear functions. This in turn generates at most m constraints and one additional variable (the upper bounding variable for $pl_{xj}(u_{xj})$); each $\rho_x(u_{xj}, u_{xk})^-$ is the maximum of at most 2 linear functions which in turn generates at most 2 constraints and an additional variable. As for the constraints for S_x, not counting the lower and upper bounds for location variables, there are at most n(n+1)/2 two-variable constraints, each for a distinct location variable pair. Thus, P_{xL}" has at most n+n+(n+1)n/2 variables and mn+3(n+1)n/2 constraints. On average, this figure is much smaller since each pl_{xj} is generally the maximum of a few linear functions and the constraints for S have few two-variable constraints. Generally, the number of constraints in P_{xL}" is much larger than the number of variables in P_{xL}", so that it is advantageous to solve the dual problem. Another way to reduce the size of P_{xL}" is to remove those linear functions in the PLC underestimates which do not affect the function's lower bounding quality significantly. For example, a heuristic to remove linear functions in a PLC function $pl_{xj}(.)$ is the following: let $[a_h, b_h]$ be the maximal interval such that linear function $l_{xjh}(u_{xj}) = pl_{xj}(u_{xj})$ for any $u_{xj} \in [a_j, b_j]$; If $(b_h - a_h) \leq \alpha$ and $|l_{xjh}(a_h) - l_{xjh}(b_h)| \leq \beta$

for some pre-specified values α and β, then we remove $l_{xjh}(.)$. We leave the topic of how to solve P_{xL}" efficiently to future study.

4.5.7. The Multicenter Lower Bounding Problems

The multicenter problem is P: Minimize $\{f(U) \mid U \in N_g^n\}$, where $f(U) = \max\{ \max\{f_j(u_j), j = 1, \ldots, n\}, f_{NN}(U) \} \mid U \in N_g^n\}$ with $f_j(u_j) = \max\{w_{ij}d(v_i, u_j), i = 1, \ldots, m\}$, and $f_{NN}(U) = \max\{v_{jk}d(u_j, u_k), j < k\}$. A subproblem is P': Minimize $\{f(U) \mid U \in S \cap N_g^n\}$, where $S = S_x \times S_y$ with S_x and S_y some polytopes defined as in Subsection 4.5.2. Unlike the multimedian problem, we cannot decompose P' into two independent problems, except some special cases. In the following, we first introduce a lower bounding problem for P' and then discuss some preprocessing procedures for reducing the size of the lower bounding problem.

4.5.7.1. The Lower Bounding Problem

First of all, we see that each $f_j(u_j)$ can be regarded as the objective function of a 1-Center problem on N_g. Let B_j^c denote the set of bottleneck points on N_g with respect to $f_j(u_j)$.

Observation 4.12. On each grid line, $f_j(u_j)$ has at most $2m^2(m+1)$ bottleneck points.

Proof. Suppose u_j is fixed to a horizontal grid line, so that u_{yj} is a constant, say, c, and $f_j(u_j)$ is the maximum of functions $\delta_x(v_{xi}, u_{xj}) + c_{ij}$, $i = 1, \ldots, m$, where $c_{ij} = |v_{yi} - c|$. Since u_{yj} is fixed, from Observation 4.9, we know that either $\delta_x(v_{xi}, u_{xj}) = \phi(u_{xj} \mid v_{xi}, vl_{[i]}, vl_{[i]}')$ or $\delta_x(v_{xi}, u_{xj}) = |v_{xi} - u_{xj}|$, so that $\delta_x(v_{xi}, u_{xj})$ consists of at most 4 linear functions. Thus, we can decompose the grid line into at most 4m intervals in each of which every $\delta_x(v_{xi}, u_{xj})$ is linear. In each such interval, there are at most $m(m+1)/2$ bottleneck points. Therefore, there are at most $2m^2(m+1)$ bottleneck points on this grid line. The above analysis is true for any grid line. Thus, the observation is true. ∎

For the given S, let $\{J_x, J_y, J_c\}$ be a partition of the new facility index set, with J_x (J_y) the set of those j's such that u_{yj} (u_{xj}) is fixed (at a grid line coordinate), and J_c the set of the rest of location variable indices. We can express P' as

$$\text{P}': \underset{U \in S \cap N_g^n}{\text{Minimize}} \max\{ \max\{f_j(u_{xj}), j \in J_x\}, \max\{f_j(u_{yj}), j \in J_y\}, \max\{f_j(u_j), j \in J_c\}, f_{NN}(U)\}.$$

From Observation 4.12, each $f_j(u_{xj})$, $j \in J_x$ is a piecewise linear function with at most $2m^2(m+1)$ break points (bottleneck points of $f_j(u_{xj})$). Thus, we can apply the procedure in Appendix B.3 to obtain the PLC supporting plane, denoted as pl_j, for each such f_j. Similarly, we can construct the PLC supporting plane for each f_j, $j \in J_y$. Also note that $f_j(u_j)$, $j \in J_c$ is the maximum of functions $\delta_x(v_{xi}, u_{xj}) + \delta_y(v_{xi}, u_{yj})$, and $f_{NN}(U)$ is the maximum of functions $\rho_x(u_{xj}, u_{xk}) + \rho_y(u_{yj}, u_{yk})$. We thus obtain the following lower bounding problem for P':

P_L': $\underset{U \in S}{\text{Minimize}}$ $\max\{ \max\{pl_j(u_{xj}), j \in J_x\}, \max\{pl_j(u_{yj}), j \in J_y\}, \max\{f_j(u_j)^-, j \in J_c\}, f_{NN}(U)^-\}$,

where $f_j(\,)^-$, $j \in J_c$, is obtained from $f_j(\,)$ by replacing each $\delta_x(.\,,.)$ and $\delta_y(.\,,.)$ with the PLC underestimates $\delta_x(.\,,.)^-$ and $\delta_y(.\,,.)^-$ as discussed in subsection 4.5.4.1; and $f_{NN}(\,)^-$ is obtained from $f_{NN}(\,)$ by replacing each $\rho_x(.\,,.)$ and $\rho_y(.\,,.)$ with their PLC underestimate $\rho_x(.\,,.)^-$ and $\rho_y(.\,,.)^-$ as discussed in Subsection 4.5.4.2. Clearly, P_L' is a linear programming problem.

As for the size of P_L', note that each pl_j is the maximum of at most $2m^2(m+1)$ linear functions which in turn generates at most $2m^2(m+1)$ constraints, each $\delta_x(v_{xi}, u_{xj})^- + \delta_y(v_{yi}, u_{yj})^-$ in $f_j(u_j)^-$, $j \in J_c$, generates at most 6 constraints, and each $\rho_x(u_{xj}, u_{xk})^- + \rho_y(u_{yj}, u_{yk})^-$ in $f_{NN}(\,)^-$ generates at most 12 constraints. For the constraints of S, not counting the lower and upper bounds for the location variables, each location variable pair is associated with at most one two-variable constraint, so that there at most $n(n+1)/2$ constraints. Thus, P_L' has at most $2m^2(m+1)(|J_x|+|J_y|) + 6m|J_c| + 13n(n+1)/2$ constraints; and at most $2n+1$ variables. In the worst-case, P_L' has $2m^2(m+1)n + 13n(n+1)/2$ constraints and $2n+1$ variables. The average figure should be much less than this worst-case figure, since each pl_i is the maximum of only a few linear functions, and S has few two-variable constraints.

4.5.7.2. Some Preprocessing Procedures

In case the size of P_L' is too large, some preprocessing is necessary. Since P_L' is to minimize the maximum of a set of linear functions, there is great potential for reducing the size of P_L' by eliminating those linear functions which never become binding or have an insignificant effect on the quality of P_L'. In the following, we give some procedures which identify some of these redundant linear functions. Let $R_j = [lb_j, rb_j] \times [bb_j, tb_j]$.

Preprocessing:

Step 1. Let $f_j^- = \min\{f_j(u_j) \mid u_j \in R_j \cap N_g\}$, $f_j^+ = \max\{f_j(u_j) \mid u_j \in R_j \cap N_g\}$ for $j = 1, \ldots, n$.

Let $LB = \max\{f_1^-, \ldots, f_n^-\}$, and $J_0 = \{j \mid f_j^+ \leq LB\}$;

Step 2. Let $J_x = J_x - J_0$, $J_y = J_y - J_0$, $J_c = J_c - J_0$ (an optimal u_j can be any point in $S \cap N_g^n$, for $j \in J_0$)

For each $j \in J_x$, let

$\alpha_{jp}^- = \text{minimum}\ \{l_p(u_{xj}) \mid u_{xj} \in [lb_j, rb_j]\}$ and $\alpha_{jp}^+ = \text{maximum}\{l_p(u_{xj}) \mid u_{xj} \in [lb_j, rb_j]\}$

for each linear function $l_p(u_{xj})$ in $pl_j(u_{xj})$.

For each $j \in J_y$, find α_{jq}^- and α_{jq}^+ for each linear function $l_q(u_{yj})$ in $pl_j(u_{yj})$;

For each $j \in J_c$, let

$\beta_{ij}^- = \text{minimum}\{w_{ij}\delta_x(v_{xi}, u_{xj})^- \mid lb_j \leq u_{xj} \leq rb_j\} + \text{minimum}\{w_{ij}\delta_y(v_{yi}, u_{yj})^- \mid bb_j \leq u_{yj} \leq tb_j\}$ and

$\beta_{ij}^+ = \text{maximum}\{w_{ij}\delta_x(v_{xi}, u_{xj})^- \mid lb_j \leq u_{xj} \leq rb_j\} + \text{maximum}\{w_{ij}\delta_y(v_{yi}, u_{yj})^- \mid bb_j \leq u_{yj} \leq tb_j\}$.

For each $j < k$, let

$\gamma_{jk}^- = \text{minimum}\{v_{jk}(\rho_x(u_{xj}, u_{xk})^- \mid (u_{xj}, u_{xk}) \in X_{jk}\} + \text{minimum}\{v_{jk}\rho_y(u_{yj}, u_{yk})^-) \mid (u_{xj}, u_{xk}) \in Y_{jk}\}$,

$\gamma_{jk}^+ = \text{maximum}\{v_{jk}(\rho_x(u_{xj}, u_{xk})^- \mid (u_{xj}, u_{xk}) \in X_{jk}\} + \text{maximum}\{v_{jk}\rho_y(u_{yj}, u_{yk})^-) \mid (u_{xj}, u_{xk}) \in Y_{jk}\}$;

Step 3. Let $L^- = \{LB, \alpha_{jp}^-, \ldots \beta_{ij}^-, \ldots \gamma_{jk}^-, \ldots\})$ and $L^+ = \{\alpha_{jp}^+, \ldots, \beta_{ij}^+, \ldots, \gamma_{jk}^+, \ldots\}$.

For any two elements a and b with $a \in L^+$ and $b \in L^-$, if $a \leq b$ then eliminate all the linear functions associated with a from P_L'.

Now, we give some analysis for this preprocessing procedure.

First of all, it is not difficult to obtain these lower bounds and upper bounds in the procedure. We can use the algorithm in Hakimi (1979) to find each f_j^- in $O(|E|n\log n)$. Since each f_j^+ corresponds to an antipodal point of N_g, we can find f_j^+ in $O(|E|m\log m)$, where m comes from the fact that in each edge there are at most m antipodal points. It is obvious that those α_{jp}^-, α_{jp}^+, β_{ij}^-, β_{ij}^+ can be easily obtained. To obtain γ_{jk}^-, one needs to solve two independent problems of finding a minimum for a PLC function on a simple polytope. To obtain γ_{jk}^+, one need to solve two independent problems of finding a maximum for a PLC function on a simple polytope with known extreme points.

Secondly, we justify the elimination measures in the procedure. It is clear that for each $j \in J^0$, the function $f_j(u_j)$ will not affect function f(U) for any solution u_j, so that we can eliminate the entire function from f(U). Each element in L^- is a lower bound for P_L' and each element in L^+ is an upper bound for some function which is part of the objective of P_L'. Thus, if the upper bound for a function is not greater than some lower bound for P_L', the function can be eliminated from P_L'.

Finally, note that there is room to reduce further the size of P_L'. For example, if a linear function in some $pl_j(.)$ has no significant effect on the quality of $pl_j(.)$, that is, removing such linear function does not decrease the minimum value of the $pl_j(.)$, then we can eliminate it from P_L''. Similar eliminations can be made to $\delta_x(.\,,.) + \delta_y(.\,,.)$ and $\rho_x(.\,,.) + \rho_y(.\,,.)$. We will leave this to future study.

4.5.7.3. An Example

Consider a three new facility instance of P defined on the network G shown in Figure 4.24 and the weights given in Example 4.17. Let P' be a subproblem with U restricted to the S given in Example 4.17 of subsection 4.5.6. For P', we have $J_x = \varnothing$, $J_y = \{1, 2\}$, and $J_c = \{3\}$. Consequently, we have

$pl_1(u_{x1}) = 48$, $pl_2(u_{x2}) = \max\{1.8144u_{x2} + 48, 8u_{x2} + 8\}$,

$f_3(u_3)^- = \max\{w_{i3}(|v_{xi} - u_{x3}| + |v_{yi} - u_{y3}|) \mid i = 1, \ldots, 20\}$,

$\rho_x(u_{x1}, u_{x2})^- = \max\{u_{x1}, u_{x2} - u_{x1}\}$, $\rho_y(u_{y1}, u_{y2})^- = 2$

$\rho_x(u_{x1}, u_{x3})^- = |u_{x1} - u_{x3}|$, $\rho_y(u_{y1}, u_{y3})^- = u_{y3}$

$\rho_x(u_{x2}, u_{x3})^- = |u_{x2} - u_{x3}|$, $\rho_y(u_{y2}, u_{y3})^- = 2 - u_{y3}$

$f_1^- = 48$; $f_2^- = 48$, $f_3^- = 24$ (so that $LB = 48$)

$f_1^+ = 60$; $f_2^+ = 72$, $f_3^+ = 50$;

After preprocessing, the linear functions associated with $pl_1(u_{x1})$ or $f_3(u_3)^-$ are completely eliminated from P_L'. We get an equivalent lower bounding problem

$$P_L': \underset{U \in S}{\text{minimize}} \max\{pl_2(u_{x2}), 10(\rho_x(u_{x1}, u_x)^- + \rho_y(u_{y1}, u_{y2})^-), 10(\rho_x(u_{x1}, u_{x2})^- + \rho_y(u_{y1}, u_{y3})^-), 10(\rho_x(u_{x2}, u_{x3})^- + \rho_y(u_{y2}, u_{y3})^-)\}.$$

Problem P_L' is then transformed into a LP problem with 8 variables and 11 constraints. The optimal objective value of P_L' is 48 which is also the optimal objective value of P'.

4.6 Summary

In this chapter, we showed that the n-fold Cartesian product of a cyclic network can be partitioned into a finite number of subsets on each of which both types of distances are linear functions. Based on this partition, we defined a special type of solution subset which is useful in a branch and bound algorithm for a general multifacility problem with objective involving a convex function of both types of distances. For the subproblems defined on such a solution subset, we introduced some lower bounding techniques based on the piecewise linearity property of the network distance. The lower bounding problems are all linearly constrained convex programs, and they become linear in objective function for the multimedian and the multicenter problems. For grid network multifacility problems, we defined similar solution subsets and lower bounding problems. Compared with the problems on general cyclic networks, the grid network specialization enables us to devise lower bounding problems with substantial improvements in approximation quality.

CHAPTER 5
SUMMARY

In this dissertation, we considered a class of network location problems – the multifacility location problems, which are known to involve distances between pairs of facilities. We developed theories and algorithms for the problems on some special cyclic networks. The results are useful for both solving the problems considered in this dissertation and for further understanding the properties of this class of location problems on cyclic networks.

In Chapter 2, we established a localization theory for the multimedian problem on multiblock networks. This localization theory enables us to localize, in polynomial time, every location variable to either a vertex or a block of the network. This result demonstrates the potential of understanding the relation between network distances and network structures.

In Chapter 3, we developed a B&B algorithm for the multimedian problem on grid networks. We gave a dominating relation which leads to a useful polynomially solvable approximation – the intersection-restricted multimedian problem. We also give several search heuristics. Numeric testing showed that the B&B algorithm can solve practical-size problems to optimality. The test showed that the approximation problem was adequate in providing a near optimal solution.

In Chapter 4, we developed some lower bounding techniques for a class of multifacility location problems. For the general case when the underlying network is a general cyclic network, we identified a partition of the solution space, defined the solution subsets and hence the subproblems a B&B algorithm should use, and introduced some piecewise linear and convex underestimates for the subproblem objectives. For the special case when the underlying network is a grid network, we made substantial improvement on the piecewise linear and convex underestimates.

There are many open topics for this class of multifacility problems on cyclic networks. Because of the presences of multiple local optimal solutions, the multifacility problem falls into the category of global optimization. Partly due to the lack of simple (e.g. polygonal) solution space representation, and partly due to the complexity of the objective functions, it is still an open question of applying known global optimization solution techniques to the multifacility problem. There may exist other forms of localization for other multifacility location problems on multiblock networks. The solution partitioning and the lower bounding techniques given in Chapter 4 maybe further generalized to grid networks under the presence of barriers of various shapes. We need to develop a complete B&B algorithm for some multifacility problems, which utilizes the methods proposed in Chapter 4. Since the multimedian problem is a relaxation of the well-known quadratic assignment problem, it is interesting to apply and to modify the lower bounding techniques for the latter problem to get better approximation algorithms for the former. Finally, there is potential in developing B&B algorithm for multifacility problems, which utilizes various known lower bounds and solution set partitioning strategies.

APPENDIX A
THE PROOFS IN CHAPTER 3

First, we establish some terminology and state some properties of the tree network multimedian problem. Let P be a MMP on a tree network T with objective function f. Let $J = \{1, ..., n\}$ be the new facility index set. Throughout this appendix, we will use some graphical examples to illustrate ideas. These examples are indicated with parenthetical references (e.g. (See Example A1)).

Definition A3.1. An adjacent movement is a triplet $<(v_s, v_t), S, X>$ denoting the process of changing a solution X of P by moving some subset $S \subseteq J$ of new facilities from their common location at vertex v_s to an adjacent vertex v_t.

We see that for the given subset S and the vertex v_s, solution X must be one of those which has their new facilities involved in S located at vertex v_s. Let $D(S, v_s) = \{X \mid x_k = v_s, k \in S\}$. Lemma A3.5 below gives a sufficient condition for such a movement to decrease the objective function value. In order to state Lemma A3.5, Lemma A3.1 and Lemma A3.2 give some simple facts on how the objective value changes as a result of an adjacent movement. Lemma A3.3 and Lemma A3.4 associate an adjacent movement with the optimality condition for the tree MMP.

Definition A3.2. For a given vertex solution X to P, an edge (v_s, v_t) of T, and a subset S of J, define a partition $\{J_s(X), J_t(X)\}$ of $J - S$ over subtrees T_s and T_t as $J_s(X) = \{k \in J - S, x_k \in T_s\}$ and $J_t(X) = \{k \in J - S, x_k \in T_t\}$ (See Example A1).

Set $J_s(X)$ consists of the indices of those new facilities remaining in subtree T_s after adjacent movement $<(v_s, v_t), S, X>$. Likewise, $J_t(X)$ consists of the indices of those new facilities which are in subtree T_t before the movement. Note that the partition is defined with respect to not only a solution X but also to a subset S of new facility indices and an edge of T.

Lemma A3.1. For a given vertex solution X of P, let X' be the solution obtained from X by an adjacent movement $<(v_s, v_t), S, X>$. Then, $f(X') - f(X) = \delta(X, S)\, d(v_s, v_t)$, where

$\delta(X, S) = \Sigma_{j\in S}\{ (\Sigma\{w_{ij}|v_i \in T_s\} + \Sigma\{v_{jk}|k \in J_s(X)\}) - (\Sigma\{w_{ij}|v_i \in T_t\} + \Sigma\{v_{jk}|k \in J_t(X)\}) \}$.

We see that $\delta(X, S)$ is a function defined with respect to an adjacent movement $<(v_s, v_t), S, X>$. For the given S, v_s, and v_t, it is a function of X with domain $D(S, v_s)$. In fact, $\delta(X, S)$ only depends on the partition $\{J_s(X), J_t(X)\}$ instead of the exact new facility locations. Therefore, for any two solutions X and Y in $D(S, v_s)$, $\delta(Y, S) = \delta(X, S)$ if $J_s(Y) = J_s(X)$ and $J_t(Y) = J_t(X)$. Furthermore, if $J_s(X)$ is a proper subset of $J_s(Y)$, then the weights v_{jk}'s with $j \in S$ and $k \in J_s(Y) - J_s(X)$, which have positive coefficients in $\delta(Y, S)$, have negative coefficients in $\delta(X, S)$. Thus,

Lemma A3.2. Let S be a given subset of new facility indices. For any X and Y in $D(S, v_s)$, if $J_s(X) \subseteq J_s(Y)$, then $\delta(X, S) \le \delta(Y, S)$.

Proof. From the above discussion, we know that

$$\delta(X, S) - \delta(Y, S) = -2\Sigma_{j\in S}\Sigma\{v_{jk} \mid k \in J_s(Y) - J_s(X)\} \le 0. \text{ (See Example A2)}$$

Now, we begin to identify when $\delta(X, S) \le 0$ given that we already know a vertex optimal solution X^* to P. Let $\{O_s^*, O_t^*\}$ be an optimal partition of J such that O_s^* (O_t^*) consists of the indices of all the new facilities located in subtree T_s (T_t) in X^*. From Kolen's optimality condition (1980) we have

Lemma A3.3. Let S be a subset of O_t^*. Let X be the solution in which every new facility k with k either in O_s^* or in S located at vertex v_s (i.e. $x_k = v_s$ if $k \in O_s^* \cup S$) and in which every new facility k with k in $O_t^* - S$ located at vertex v_t (i.e. $x_k = v_t$ if $k \in O_t^* - S$). Then, $\delta(X, S) \le 0$.

Proof. Let X' be the solution with every new facility located on v_s. From Kolen's optimality condition of the tree multimedian problem, we know that $\delta(X', O_t^*) = \text{minimum}\{\delta(X', L) | L \subseteq J\}$. That is, O_t^* is one of those new facility index subsets such that moving the new facilities in such a subset from v_s to v_t decreases the objective function the most. Since this adjacent movement can be accomplished by two adjacent movements one moving new facilities in S from v_s to v_t and the other moving new facilities in O_t^*-S from v_s to v_t, we know that the objective function change

$\delta(X', O_t^*)\, d(v_s, v_t)$ for the first movement must be the sum of the objective function changes $\delta(X', O_t^* - S)\, d(v_s, v_t)$ and $\delta(X, S)\, d(v_s, v_t)$ for the latter two movements. Hence, $\delta(X', O_t^*) = \delta(X', O_t^* - S) + \delta(X, S)$. Since $\delta(X', O_t^*) \le \delta(X', O_t^* - S)$ from its minimum property, we know that $\delta(X, S)$ must be non-positive. ∎

Note that the partitions of $J - S$ (over T_s and T_t) for both solutions X and X^* in Lemma A3.3 are identical (i.e. $J_s(X) = J_s(X^*)$ and $J_t(X) = J_t(X^*)$). Since, for any solution Y in $D(S, v_s)$, if $J_s(X) = J_s(Y)$ and $J_t(X) = J_t(Y)$, then $\delta(X, S) = \delta(Y, S)$, we have the following extension of Lemma A3.3.

Lemma A3.4. Let S be a subset of new facility indices and X^* a vertex-optimal solution to P. For any solution X in $D(S, v_s)$, if $J_s(X) = J_s(X^*)$ and $J_t(X) = J_t(X^*)$, then $\delta(X, S) \le 0$.

The above lemma says that $\delta(X, S)$ is non-positive if there exists an optimal solution X^* to P such that x_k^* and x_k belong to the same subtree for every k not in S. The following lemma shows that the conditions in Lemma A3.4 can be relaxed due to the result in Lemma A3.2.

Lemma A3.5. Let S be a given subset of new facility indices. For any $X \in D(S, v_s)$, suppose there exists a vertex optimal solution X^* such that

c1) for every $k \in S$, $x_k^* \in T_t$;

c2) for every $k \notin S$, if $x_k^* \in T_t$ then $x_k \in T_t$;

Then $\delta(X, S) \le 0$.

Proof. The difference between the conditions in this lemma and the conditions in Lemma A3.4 is that the conditions in Lemma A3.4 imply condition c.2 but not vice versa. In other words, for any new facility index k with x_k^* in T_s, this lemma does not require x_k to be in T_s while Lemma A3.4 does. Thus, we know that $J_s(X) \subseteq J_s(X^*)$. If $J_s(X) = J_s(X^*)$, then the conditions in Lemma A3.4 are satisfied so that $\delta(X, S) \le 0$. Otherwise, let Y be any solution in $D(S, v_s)$ such that $J_s(Y) = J_s(X^*)$ and $J_t(Y) = J_t(X^*)$. From Lemma A3.4, we know that $\delta(Y, S) \le 0$. Since solutions X and Y satisfy the conditions in Lemma A3.2 (i.e. $J_s(X) \subseteq J_s(Y)$), therefore $\delta(X, S) \le \delta(Y, S) \le 0$. ∎

Now, we can prove Property 3.1.

Property 3.1. Let P be a MMP on a path network T. For an optimal solution X^* and another arbitrary solution X to P, any λ-combination X' of X and X^* ordered like X^* dominates X.

Proof. For ease of exposition, we assume that X', X, and X^* are all vertex solutions (otherwise, we introduce dummy vertices of zero weights). Let $L = L(X \mid X^*)$ and $R = R(X \mid X^*)$ be the sets in Definition 3.2. Finally, let J be the new facility index set. We prove Property 3.1 by showing that X' can be obtained from X by performing adjacent movements finitely many times. Each movement moves a subset of new facilities an edge length closer to their corresponding locations in X' without causing the objective function f to increase.

Now, we define the adjacent movements. The *r*th adjacent movement is a triplet $<(v_s, v_t), S, X^r>$ where X^r is the solution before the movement and (v_s, v_t) is the edge along which the subset S of new facilities is moved. Subset S is either one of the following two sets:

$S_L = \{j \mid x_j^r = \text{minimum } \{x_k^r \mid x_k^r < x_k'\}\}$ and $S_R = \{j \mid x_j^r = \text{maximum } \{x_k^r \mid x_k^r > x_k'\}\}$.

(See Example A3 for S_L and S_R)

We see that set S_L contains the "leftmost" among those new facilities which are "to the left" of their target locations in X'. Set S_R contains those "rightmost". If S_L and S_R are both empty, then X^r is X' so that the process stops. Otherwise, the process chooses either S_L or S_R as set S and obtains X^{r+1} by moving the new facilities in S from their common location in X^r to the adjacent vertex closer to every x_j' with $j \in S$. We see that the moving process moves no facilities which are already at their target locations in X' and each iteration moves at least one new facility an edge closer to its target. Therefore, we can get X' by this moving process in finitely many steps. What remains is to show that $f(X^{r+1}) - f(X^r) \leq 0$.

We only need to prove this inequality when $S = S_L$, since the proof is symmetric when $S = S_R$. With (v_s, v_t) the edge along which the new facilities in S are moved (from v_s to v_t), let T_s and T_t, $v_s \in T_s$ and $v_t \in T_t$, be the subtree pair separated at edge (v_s, v_t). From Lemma A3.1, we know that $f(X^{r+1}) - f(X^r) = \delta(X^r, S)\, d(v_s, v_t)$. Hence, it is sufficient to show that $\delta(X^r, S) \leq 0$. Lemma A3.5 gives a set of sufficient conditions for $\delta(X^r, S) \leq 0$. That is,

c.1. for every $k \in S$, $x_k^* \geq v_t$ (i.e. $x_k^* \in T_t$);

c.2. for every $k \notin S$, if $x_k^* \geq v_t$ (i.e. if $x_k^* \in T_t$) then $x_k^r \geq v_t$;

Condition c.1 is always true (Since for each $k \in S$, $x_k^r = v_s < x_k^*$, thus, $x_k^* \geq v_t$. See Example A3). Now, we consider what kind of new facility k can violate Condition c.2, i.e. $x_k^* \geq v_t$ but $x_k^r < v_t$. First of all, k cannot be in R, since, with k in R, x_k^r is always to the right of target location x_k' and hence is always to the right of x_k^*, so that if $x_k^* \geq v_t$ we must have $x_k^r \geq v_t$. Thus, such a k must be in L – S. Now, L – S can be partitioned into two sets according to the subtrees to which the new facilities in L – S belong. Let $\{L_s, L_t\}$ denote this partition, $L_s = \{k \in L - S \mid x_k^r \leq v_s\}$ and $L_t = \{k \in L - S \mid x_k^r \geq v_t\}$ (See Example A4 for examples of L_s and L_t). We see that Condition c.2 is violated only if L_s contains some new facilities which have their respective optimal locations in X^* located in subtree T_t. The rest of the proof then is to show that $\delta(X^r, S)$ is still nonpositive when such a case is true. We will prove this by first defining another solution Y to P and an adjacent movement $<(v_a, v_b), S, Y>$ along some edge (v_a, v_b) defined later. Then, we will show that $\delta(Y, S) \leq 0$. Finally, we will show that $\delta(X^r, S) \leq \delta(Y, S)$.

Solution Y is derived from X^r by reassigning the locations of some new facilities as follows. Let $v_b = \min\{x_k^* \mid k \in S\}$ and let v_a be the adjacent vertex with $v_a < v_b$. Let T_a and T_b be the subtrees separated at edge (v_a, v_b). Then, Y is obtained from X^r by relocating new facilities in S to vertex v_a, relocating each new facility k in L_t to x_k^*, and letting the remaining new facilities be at the same locations as in X^r. That is,

$$y_k = \begin{cases} v_a & \text{if } k \in S \\ x_k^* & \text{if } k \in L_t \\ x_k^r & \text{if } k \in R \cup L_s. \end{cases} \qquad \text{(See Example A5 for examples of Y, } v_a\text{, and } v_b\text{)}$$

Now we show that $\delta(Y, S) \leq 0$ by showing that this adjacent movement corresponding to $\delta(Y, S)$ satisfies Condition c.1 and c.2. That is,

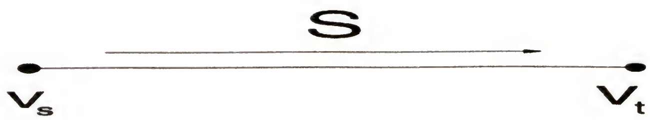
c.1. for any $k \in S$, $x_k^* \geq v_b$;

c.2. for any $k \notin S$, if $x_k^* \geq v_b$ then $y_k \geq v_b$;

Since $v_b = \min\{x_k^* \mid k \in S\}$, c.1 is true. To show c.2, we first observe that for any $j \in L_s$, we have $x_j^* < v_b$ (See Example A6 for an example showing $x_j^* < v_b$ for every $j \in L_s$) (The following is the proof of this observation: Recall that in iteration r, set S consists of the "leftmost" among those new facilities, in L, which have not reached their respective target locations in X'. On the other hand, from the definition of L_s we know that $x_j^r \le v_s$ for every j in L_s so that every new facility j in L_s is to the left of all the new facilities in S. This shows that each new facility j in L_s must have reached its target location. That is, $x_j^r = x_j'$ for every $j \in L_s$. Thus, for any $j \in L_s$ and any $k \in S$, we have $x_j' = x_j^r < v_t \le x_k'$, where the last inequality is derived from the fact that $x_k^r = v_s$ and $x_k' > x_k^r$ for every $k \in S$. Since X' is ordered like X^*, we have $x_j^* < x_k^*$. Hence, $x_j^* < v_b = \min\{x_k^* \mid k \in S\}$). Thus, if there is an index k such that $k \notin S$ and $x_k^* \ge v_b$, then k must be in $R \cup L_t$. If $k \in R$, then since x_k^r is always to the right of x_k^* and since $y_k = x_k^r$, we know that $y_k \ge v_b$. If $k \in L_t$, then, since $y_k = x_k^*$, $x_k^* \ge v_b$ implies $y_k \ge v_b$. All together, we know that condition c.2 is satisfied.

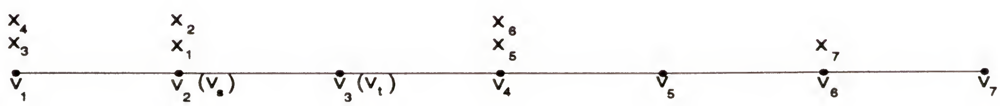
To conclude, we need to show $\delta(X^r, S) \le \delta(Y, S)$. From Lemma A3.1 we know that both $\delta(X^r, S)$ and $\delta(Y, S)$ are sums of signed weights $\pm w_{ij}$, for every vertex v_i and every new facility $j \in S$, and signed weights $\pm v_{jk}$, for every new facility pair $j \in S$, $k \notin S$. For each signed weight sw_{ik} (sv_{jk}) in $\delta(X^r, S)$ there is a corresponding signed weight sw_{ik}' (sv_{jk}') in $\delta(Y, S)$, which is either identical to sw_{ik} (sv_{jk}) or differs by a sign. Thus, $\delta(X^r, S) \le \delta(Y, S)$ if we can show that for each positively signed weight in $\delta(X^t, S)$ the corresponding signed weight in $\delta(Y, S)$ remains positively signed. First, we see that sw_{ij} in $\delta(X^r, S)$ (sw_{ij}' in $\delta(Y, S)$) is positively signed if and only if $v_i \le v_s$ ($v_i \le v_a$). Since $v_s < v_a$, any vertex v_i with $v_i \le v_s$ has $v_i \le v_a$, so that every positively signed sw_{ij} in $\delta(X^r, S)$ is also positively signed in $\delta(Y, S)$. Secondly, sv_{jk} in $\delta(X^r, S)$ (sv_{jk}' in $\delta(Y, S)$) is positively signed if and only if $x_k^r \le v_s$ ($y_k \le v_a$). For every k such that $x_k^r \le v_s$, $k \notin S$, k is either in R or in L_s so that from the definition of Y, $y_k = x_k^r \le v_s < v_a$. This means that every positively signed sv_{jk} in $\delta(X^r, S)$ is also positively signed in $\delta(Y, S)$. All together, we know that every positively signed weight in $\delta(X^r, S)$ remains positively signed in $\delta(Y, S)$. Hence, $\delta(X^r, S) \le \delta(Y, S)$. Thus, the property is true. ∎

Examples in the Appendix

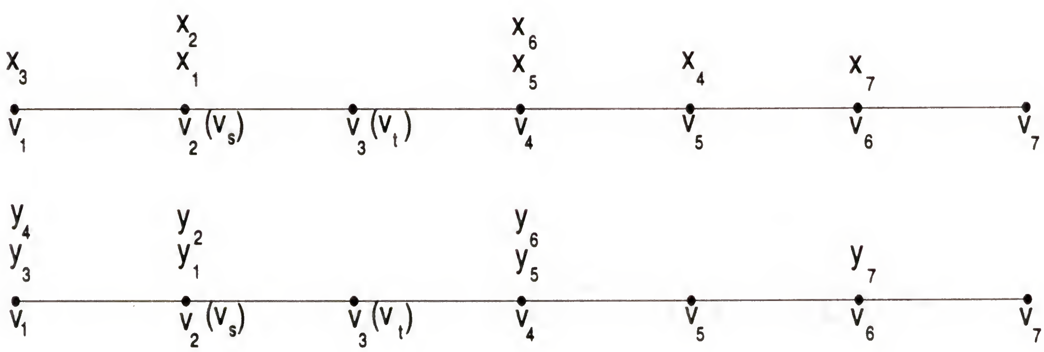


Ex.A1 (A Partition of J-S)

For the given solution X shown below and S = {1, 2}, $v_s = v_2$, and $v_t = v_3$, we have $J_s(X)$ = {3, 4}, $J_t(X)$ = {5, 6, 7}.



Ex. A2. Let X and Y be the solutions shown below.



With S = {1, 2}, we have $J_s(X)$ = {3}, $J_t(X)$ = {4, 5, 6, 7}, $J_s(Y)$ = {3, 4}, and $J_t(Y)$ = {5, 6, 7}.

Thus,

$$J_s(X) \subseteq J_s(Y),$$

$$\delta(X, S) = \Sigma_{j\in S}[\Sigma_{i=1,2}w_{ij} + \Sigma\{v_{kj}|\ k\in J_s(X)\} - \Sigma_{i=3,\dots,7}w_{ij} - \Sigma\{v_{kj}|k\in v_{kj}\}$$
$$= \Sigma_{j\in S}[w_{1j} + w_{2j} + w_{3j} - \Sigma_{i=3,\dots,7}w_{ij} - v_{4j} - v_{5j} - v_{6j} - v_{7j}]$$

and

$$\delta(Y, S) = \Sigma_{j\in S}[\ \Sigma_{i=1,2}w_{ij} + \Sigma\{v_{kj}|k\in J_s(Y)\} - \Sigma_{i=3,\ldots,7}w_{ij} - \Sigma\{v_{kj}|k\in J_t(Y)]$$
$$= \Sigma_{j\in S}[w_{1j} + w_{2j} + w_{3j} + v_{4j} - \Sigma_{i=3,\ldots,7}w_{ij} - v_{5j} - v_{6j} - v_{7j}]$$

so that $$\delta(X, S) - \delta(Y, S) = -2(v_{41} + v_{42}).$$

Ex. A3. For the optimal solution X^*, an intermediate solution X^r, and the target solution X' respectively given in Figure A.1(a), (b), and (c) below, we have $S_L = \{1, 2\}$ and $S_R = \{5\}$. By selecting $S = S_L$ we then have $v_s = v_2$ and $v_t = v_3$.

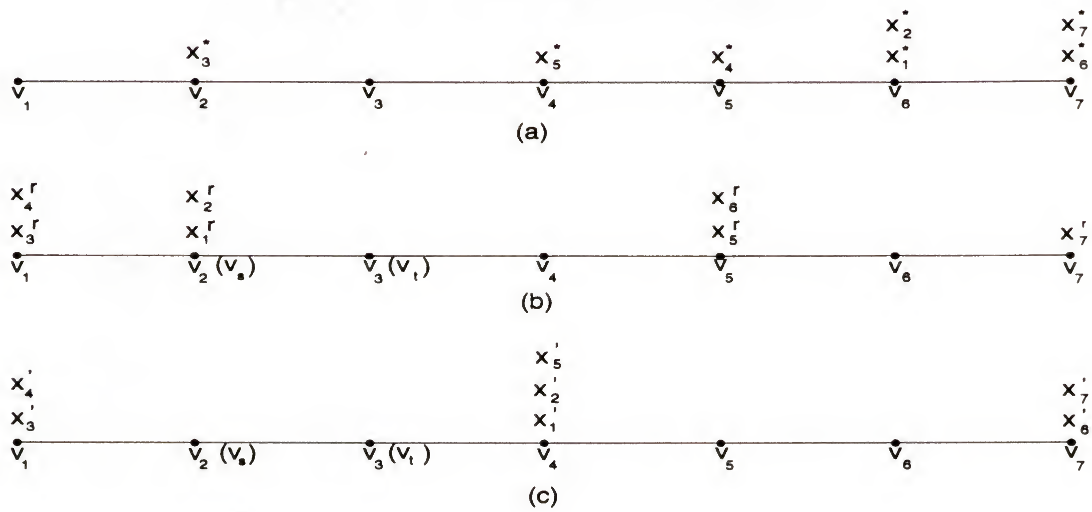


Figure A.1 X^*, X^r, and X'

Ex. A4. With the S, X^*, X', and X^r given in Figure A.1, we have $L_s = \{3, 4\}$, $L_t = \{6, 7\}$, and $R = \{5\}$. Condition c.2 is violated since for the new facility 4 in L_s, $x_4^r < v_t$ but $x_4^* > v_t$.

Ex. A5 (Constructing Solution Y)

For the X^r, X^*, X', S, L_s, L_t, and R given in the above example, we see that $v_b = v_6$ so that $v_a = v_5$. From the rules of constructing Y, we have

$y_1 = v_a = v_5$, $y_2 = v_a = v_5$, since $S = \{1, 2\}$,
$y_6 = x_6^* = v_7$, $y_7 = x_7^* = v_7$, since $L_t = \{6, 7\}$, and
$y_3 = x_3^r = v_1$, $y_4 = x_4^r = v_4$, $y_5 = x_5^r = v_5$, since $L_s = \{3, 4\}$ and $R = \{5\}$.

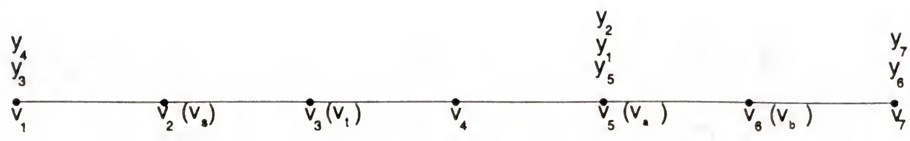


Figure A.2 Constructing Y

Thus, $Y = (v_5, v_5, v_1, v_1, v_5, v_7, v_7)$ as shown in Figure A.2. The adjacent movement is to move new facilities in $S = \{1, 2\}$ from v_a to v_b.

Ex. A6. (Condition c.2)

Consider the figures given in Ex. 3 and Ex. 5 and the subsets of indices given in Ex. A3 and Ex. A4. Here $L_s = \{3, 4\}$. We see that $x_4^* < v_b = v_6$. This is because $x_4' < v_t <= x_k'$ for every k in $S = \{1, 2\}$, so that, since X' is ordered like X^*, we have $x_4^* < x_k^*$ for every k in S.

APPENDIX B
THE PROOFS IN CHAPTER 4

Appendix B.0

Property 4.0. If function c in P is a non-decreasing function, then there is an optimal solution to P with every new facility located either on a vertex or in an edge which is a shortest path between its two end points.

Proof. Call an edge which is not a shortest path between its two end-points a r-edge. It is sufficient to show that for any given r-edge we can relocate, without increasing the objective value, all the new facilities in its interior to some locations which are not in the interior of any r-edge.

Suppose that r-edge e = (u, w) contains $x_1, \ldots, x_p$ in e^-, the set of interior points of e, and $x_1, \ldots, x_p$ are in the order $t(x_j) < t(x_{j+1})$, j = 1, …, p−1. Let $L(e)$ denote the length of e, $P(u, \ldots, w)$ a shortest path from u to w with length L. Let q be the largest index in $\{1, \ldots, p\}$ such that $t(x_q) \le L$. Finally, let Y be the solution derived from X by relocating the new facilities in $\{1, \ldots, p\}$ to the points of path P such that $d(y_j, u) = t(x_j)$, for j = 1, …, q, and $y_j = w$ for j = q+1, …, p. We known that

$$L - d(u, y_j) \le L(e) - t(x_j) \text{ and } d(y_j, u) \le t(x_j) \text{ for } j \in \{1, \ldots, p\}. \tag{A.1}$$

It is thus sufficient to show that D(Y) ≤ D(X), since c is non-decreasing, and since from X to Y we relocate at least one new facility from an interior point of some r-edge to a location which is not in the interior of any r-edge, so that by finite many such operations we can find a solution with no new facility located in the interior of any r-edge.

First, the following two cases show that

$$d(z, y_j) \le d(z, x_j), \text{ for every } j \in \{1, \ldots, p\}, \text{ and any } z \notin e^- \tag{A.2}$$

Case i. z is not in the interior of path $P(u, \ldots, w)$

We know that $d(z, y_j) = \min\{d(u, z) + d(u, y_j), d(w, z) + L - d(u, y_j)\}$. From (A.1), we have $d(z, y_j) \le \min\{d(u, z) + t(x_j), d(w, z) + L(e) - t(x_j)\} = d(z, x_j)$;

Case ii. z is in the interior of path $P(u, \ldots, w)$

Now, $d(z, y_j) = |d(u, z) - d(u, y_j)|$ and $d(z, x_j) = \min\{d(u, z) + t(x_j), L - d(u, z) + L(e) - t(x_j)\}$.

Case ii.a. $d(u, y_j) \leq d(u, z)$

In this case, $d(z, y_j) = d(u, z) - d(u, y_j) = d(u, z) - t(x_j)$. From the last equality, we have $d(z, y_j) \leq d(u, z) + t(x_j)$. Since $d(u, z) \leq L$, thus $d(z, y_j) \leq L - t(x_j) \leq L - d(u, z) + L(e) - t(x_j)$. Therefore, $d(z, y_j) \leq \min\{d(u, z) + t(x_j), L - d(u, z) + L(e) - t(x_j)\} = d(z, x_j)$;

Case ii.b. $d(u, y_j) > d(u, z)$

In this case, $d(z, y_j) = d(u, y_j) - d(u, z)$. Thus $d(z, y_j) \leq t(x_j) - d(u, z) < d(u, z) + t(x_j)$ and $d(z, y_j) \leq L - d(u, z) \leq L - d(u, z) + L(e) - t(x_j)$. Similar to Case ii.a, we have $d(z, y_j) \leq d(z, x_j)$;

Inequality (A.2) implies the following. First, since no v_i is in e^-, thus

$$d(v_i, y_j) \leq d(v_i, x_j) \text{ for every } j \in \{1, \ldots, p\} \text{ and for every } i. \tag{A.3}$$

Secondly, for every k such that $x_k \notin e^-$, $d(y_j, x_k) \leq d(x_j, x_k)$. Thus, since $y_k = x_k$, we have

$$d(y_j, y_k) = d(y_j, x_k) \leq d(x_j, x_k),\ j \in \{1, \ldots, p\},\ k \in \{k \mid x_k \notin e^-\}. \tag{A.4}$$

Now we show that

$$d(y_j, y_k) \leq d(x_j, x_k) \text{ for any } j, k \in \{1, \ldots, p\}, j < k. \tag{A.5}$$

We know that $d(y_j, y_k) = d(u, y_k) - d(u, y_j)$ and $d(x_j, x_k) = \min\{t(x_k) - t(x_j), t(x_j) + L + L(e) - t(x_k)\}$. If $d(x_j, x_k) = t(x_j) + L + L(e) - t(x_k)$, then, since a shortest path between x_j and x_k contains the entire path $P(u, \ldots, w)$, it is obvious that $d(y_j, y_k) \leq d(x_j, x_k)$. Now, with $d(x_j, x_k) = t(x_k) - t(x_j)$, we consider three cases:

Case iii. $j > q$ and $k > q$

Now $d(u, y_j) = L$ and $d(u, y_k) = L$ so that $d(y_j, y_k) = 0 \leq d(x_j, x_k)$;

Case iv. $j \leq q$ and $k \leq q$

Now, $d(u, y_j) = t(x_j)$ and $d(u, y_k) = t(x_k)$, so that $d(y_j, y_k) = t(x_k) - t(x_j) = d(x_j, x_k)$.

Case v. $j \leq q$ and $k > q$

Now, $d(u, y_j) = t(x_j)$ and $d(u, y_k) = L < t(x_k)$, so that $d(y_j, y_k) = L - t(x_j) < d(x_j, x_k)$.

All together, we know that (A.5) is true.

The distances not considered so far are those $d(v_i, y_j)$'s for every $j > p$ and $d(y_j, y_k)$'s for j, k $> p$. But since they all involve location variables which have the same values in both X and Y, we know that

$$d(v_i, y_j) = d(v_i, x_j), \text{ for every i and every } j > p \tag{A.6}$$

and

$$d(y_j, y_k) = d(x_j, x_k), \text{ for every } j, k > p \tag{A.7}$$

Equalities (A.6), (A.7) and inequalities (A.1), ..., (A.4) cover all the distances in D( ), thus, we know that $D(Y) \leq D(X)$. ▮

<u>Appendix B.1</u>

Here, we give an algorithm for the problem of "Adding a Constraint" as defined in Subsection 4.3.4. For the given $S^0 \subset E^n$, let $\mathcal{C}^0$ be the set of binding constraints for S^0. Let M^0 be an n×n indicator matrix such that element $m_{jk}{}^0 = 0$ for all $j \geq k$, $m_{jk}{}^0 = 1$ if there is a half-plane $a_1x_j + a_2x_k \leq b$ in $\mathcal{C}^0$ and $m_{jk}{}^0 = 0$ otherwise, for $1 \leq j < k \leq n$. Let $\alpha_j{}^0 = \min\{x_j | X \in S^0\}$ and let $\beta_j{}^0 = \max\{x_j | X \in S^0\}$, $j = 1, \ldots, n$. Without loss of generality, suppose every $\alpha_j{}^0$ and $\beta_j{}^0$ are known. Hence,

$S^0 = \{X \in E^n | \alpha_j{}^0 \leq x_j \leq \beta_j{}^0, j = 1, \ldots, n, \text{ and } a_1x_j + a_2x_k \leq b \text{ for each } (j, k) \text{ such that } m_{jk}{}^0 = 1\}$.

For set S, let M, $\mathcal{C}$, α_j, and β_j be the counterparts of M^0, $\mathcal{C}^0$, $\alpha_j{}^0$, $\beta_j{}^0$ respectively. The algorithm either constructs M, $\mathcal{C}$, and every α_j and β_j, or concludes that S is empty. The algorithm starts with $M = M^0$. The addition of the new constraint may directly change the range of a variable, say x_j (or 2 variables, say x_j and x_k) associated with the constraint. If the range of x_j is changed, then the non-zero entries, in M, in the same row and the same column corresponding to x_j are marked negative. This is because every such entry corresponds to a two-variable constraint involving x_j and another variable. When the range of x_j is changed, the range of the other variable may be affected. The algorithm then proceeds to check and change the range of other variable. As a result, more non-zero entries are marked negative. The algorithm repeats this process until no entries in M are negative. The algorithm also detects and eliminate redundant constraints.

<u>An Algorithm for Constructing M</u>:

Step 0: Let $M = M^0$, $C = C^0$, $\alpha_j = \alpha_j^0$, and $\beta_j = \beta_j^0$, $j = 1, \ldots, n$;
Step 1:
  1.1: If the new constraint is a single-variable half plane $x_p \le b$ for some p, then
    Begin
      if $b < \alpha_p$, then (S is empty) go to Step 3;
      if $b \ge \beta_p$ then ($x_p \le b$ is redundant) go to Step 4;
      Else
        Begin
          let $\beta_p = b$, $m_{pk} = -|m_{pk}|$ for every $k > p$;
          $m_{jp} = -|m_{jp}|$ for every $j < p$;
          go to Step 2;
        End;
    End;
  1.2: If the new constraint is a single half plane $x_p \ge b$ for some p, then
    Begin
      if $\beta_p < b$, then (S is empty) go to Step 3;
      if $b < \alpha_p$ then ($x_p \ge b$ is redundant), go to Step 4;
      Else
        Begin
          let $\beta_p = b$, $m_{pk} = -|m_{pk}|$ for every $k > p$;
          $m_{jp} = -|m_{jp}|$ for every $j < p$;
          go to Step 2;
        End;
    End;
  1.3: If the new constraint is a two-variable half-plane $H_{pq} = \{(x_p, x_q) \mid a_1x_p + a_2x_q \le b\}$ for some p and q, then
      let $m_{pq} = -1$, $C = C \cup \{a_1x_p + a_2x_q \le b\}$, and go to Step 2;
Step 2:
  If there are no negative elements in M, then go to Step 4;
  Else
    Begin
      From M, choose arbitrarily an element, say m_{st}, which equals to -1 and let $m_{st} = 1$;
      If the rectangle $T_{st} = \{(x_s, x_t) \mid \alpha_s \le x_s \le \beta_s, \alpha_t \le x_t \le \beta_t\}$ is contained in the half-plane $H_{st} = \{(x_s, x_t) \mid a_1x_s + a_2x_t \le b\}$, then ($H_{st}$ is redundant)
      Begin
        $C = C - \{H_{st}\}$,
        $m_{st} = 0$, and
        go to the beginning of Step 2;
    End;
      If R_{st} ($= T_{st} \cap H_{st}$) $= \varnothing$ then (S is empty) go to Step 3;
      Else
    Begin
      $\alpha_s' = \min\{x_s \mid (x_s, x_t) \in R_{st}\}$, $\beta_s' = \max\{x_s \mid (x_s, x_t) \in R_{st}\}$,
      $\alpha_t' = \min\{x_t \mid (x_s, x_t) \in R_{st}\}$, $\beta_t' = \max\{x_t \mid (x_s, x_t) \in R_{st}\}$,
      If $[\alpha_s', \beta_s'] = [\alpha_s, \beta_s]$ and $[\alpha_t', \beta_t'] = [\alpha_t, \beta_t]$ then go to (the begin of) Step 2 (the range of the two variables are not changed in this iteration);
      Else
      Begin
        If $\alpha_s < \alpha_s'$ and/or $\beta_s' < \beta_s$ then
          Begin

let $m_{sk} = -|m_{sk}|$ for every $k > s$,
$m_{js} = -|m_{js}|$ for every $j < s$;
If $\alpha_t < \alpha_t'$ and/or $\beta_t' < \beta_t$ then
Begin
let $m_{tk} = -|m_{tk}|$ for every $k > t$
$m_{jt} = -|m_{jt}|$ for every $j < t$;
End;
Let $m_{st} = 1$;
Let $\alpha_s = \alpha_s'$, $\beta_s = \beta_s'$, $\alpha_t = \alpha_t'$, and $\beta_t = \beta_t'$;
End;
End;
End
Go to the beginning of Step 2;
End;
Step 3: Mark that S is empty;
Step 4: Stop;

<u>Example A4.1</u>. Consider identifying the set of non-redundant constraints for S which is derived by adding new constraint $x_2 + x_3 \le 2$ to $S^0 = \{(x_1, x_2, x_3)| \ x_1 + x_2 \le 3, x_1 + x_3 \ge 3, 0 \le x_1 \le 3, 0 \le x_2 \le 3, 0 \le x_3 \le 3\}$. The initial state after Step 0 and 1 are

M^0	1	2	3
1	0	1	1
2	0	0	0
3	0	0	0

	α_j^0	β_j^0
1	0	3
2	0	3
3	0	3

M	1	2	3
1	0	1	1
2	0	0	-1
3	0	0	0

	α_j	β_j
1	0	3
2	0	3
3	0	3

There are three iterations in Step 2. The states of M and α_j's, β_j's are listed below:

Iteration 1 (s = 2, t = 3)

M	1	2	3
1	0	-1	-1
2	0	0	1
3	0	0	0

	α_j	β_j
1	0	3
2	0	2
3	0	2

Iteration 2 (s = 1, t = 2)

M	1	2	3
1	0	1	-1
2	0	0	1
3	0	0	0

	α_j	β_j
1	0	3
2	0	2
3	0	2

Iteration 3 (s = 1, t = 3)

M	1	2	3
1	0	-1	1
2	0	0	1
3	0	0	0

	α_j	β_j
1	1	3
2	0	2
3	0	2

Iteration 4 (s = 1, t = 2)

M	1	2	3
1	0	1	1
2	0	0	1
3	0	0	0

	α_j	β_j
1	1	3
2	0	2
3	0	2

Appendix B.2

Now, we state a geometry property in E^3. Property 4.1 is a special case of this property. Again, let $\Delta(a, b, c)$ denote the convex hull in E^p spanned by three linearly independent points a, b, and c in E^p. Let $p_i = (p_{xi}, p_{yi})$, i = 1, …, 4, be the four extreme points of a quadrilateral R in space E^2, with p_1 and p_3 (p_2 and p_4) diagonal to each other. For i = 1, …, 4, let P_1 be a point in E^3 such that $P_i = (p_{xi}, p_{yi}, z_i)$ for some real number z_i. Let C be the convex hull spanned by points $P_1, \ldots, P_4$. Let $l_1(x, y)$, $l_2(x, y)$, $l_3(x, y)$, and $l_4(x, y)$ be the algebraic representation of the linear planes containing $\Delta(P_2, P_1, P_4)$, $\Delta(P_1, P_2, P_3)$, $\Delta(P_2, P_3, P_4)$, and $\Delta(P_1, P_4, P_3)$ respectively. Let $pl_1(x, y)$ ($pl_2(x, y)$) be the algebraic representation of the 2-piecewise linear surface s_{13} (s_{24}) which consists of $\Delta(P_2, P_1, P_4)$ and $\Delta(P_2, P_3, P_4)$ ($\Delta(P_1, P_2, P_3)$ and $\Delta(P_1, P_4, P_3)$).

Property A4.1. Either

(a) $pl_1(x, y) = \max\{l_1(x, y), l_3(x, y)\}$, and/or (b) $pl_2(x, y) = \max\{l_2(x, y), l_4(x, y)\}\}$.

In particular, if $P_1, \ldots, P_4$ are not in some linear plane, then exactly one of (a) and (b) is true.

Proof. Note that $R = \Delta(p_2, p_1, p_4) \cup \Delta(p_2, p_3, p_4)$ and $R = \Delta(p_1, p_2, p_3) \cup \Delta(p_1, p_4, p_3)$.

From the definition,

$$pl_1(x, y) = \begin{cases} l_1(x, y) & \text{if } (x, y) \in \Delta(p_2, p_1, p_4) \\ l_3(x, y) & \text{if } (x, y) \in \Delta(p_2, p_3, p_4), \end{cases} \quad \text{and} \quad pl_2(x, y) = \begin{cases} l_2(x, y) & \text{if } (x, y) \in \Delta(p_1, p_2, p_3) \\ l_4(x, y) & \text{if } (x, y) \in \Delta(p_1, p_4, p_3). \end{cases}$$

If $P_1, \ldots, P_4$ are in a some linear plane, all $pl_j(x, y)$'s and $s_{pq}(x, y)$'s are identical. In this case, the conclusion is obviously true.

Now, suppose that $P_1, \ldots, P_4$ are not in the same linear plane. Then, the convex hull C of $P_1, \ldots, P_4$ is a polytope in E^3 consisting of four faces corresponding to the four triangles, or equivalently, the surfaces of C consist of surfaces s_{13} and s_{24}. The assumption that $p_1, \ldots, p_4$ are extreme points of a quadrilateral implies that (i) any three of the four points $p_1, \ldots, p_4$ are linearly independent; and (ii) any three of the four points $P_1, \ldots, P_4$ are linearly independent; These two conclusions further imply that C cannot have a pyramid shape. Convex hull C then only has two other possible shapes which are demonstrated in Figure 4.14a and 4.14b. We see that for the first case, s_{24} has a rooftop shape and s_{13} has a v-shape; in the other case, s_{24} has a v-shape and s_{13} has a rooftop shape. For the first case, $l_1(x, y) \geq l_3(x, y)$ for any $(x, y) \in \Delta(p_2, p_1, p_4)$ and $l_1(x, y) \leq l_3(x, y)$ for any $(x, y) \in \Delta(p_2, p_3, p_4)$. Thus, $pl_1(x, y) = \max\{l_1(x, y), l_3(x, y)\}$ for this case. For the second case, $l_2(x, y) \geq l_4(x, y)$ for any $(x, y) \in \Delta(p_1, p_2, p_3)$ and $l_2(x, y) \leq l_4(x, y)$ for any $(x, y) \in \Delta(p_1, p_4, p_3)$. Thus, $pl_2(x, y) = \max\{l_2(x, y), l_4(x, y)\}$ for this case. ▮

We see that the quadrilateral R_{jk}' in Property 4.1 is an instance of the R here, and points P_1, ..., P_4 in Property 4.1 are also the instances of the P_1, ..., P_4 here, with $z_i = d(p_i)$. Thus, the case stated in Property 4.1 is a special case of Property A4.1.

Appendix B.3

Here, we give an algorithm for constructing the PLC y-dimension supporting plane for a set of points in E^2. Let the set of points be $S = \{(x_1, y_1), \ldots, (x_p, y_p)\}$. We assume that $x_i \leq x_{i+1}$, for i = 1, ..., p−1. Let S' be the subset of S such that for each (x_i, y_i) in S' there is point (x_h, y_h) such that $y_h < y_i$ and $x_h = x_i$. Clearly, the PLC y-coordinate supporting plane for the points in S' is the same as that for the points in S. Let *PL* denote the set of linear functions which constitute the PLC y-dimension supporting plane for the points in S.

Algorithm:

Step 1: Let M = S';

Step 2: If M is empty, then stop;

Otherwise, construct the linear y-dimension supporting plane $l(x)$ for $(x_{(1)}, y_{(1)})$ and $(x_{(2)}, y_{(2)})$ where $x_{(1)}$ and $x_{(2)}$ are smallest and the second smallest x-coordinates in M.

Step 3: If $l(x_i) \leq y_i$ for every (x_i, y_i) in S', then (a) include $l(x)$ in *PL*; (b) remove $(x_{(1)}, y_{(1)})$ from M; (c) Go to Step 2;

Otherwise, remove $(x_{(2)}, y_{(2)})$ from M and go to Step 2;

In the worst case, the algorithm constructs a linear supporting plane for every pair of points. It takes O(p) to examine whether every point is above a plane. The algorithm is $O(p^3)$.

Appendix B.4

In subsection 4.3.3, we proposed using a 2-piecewise linear function $pl_i(z_1, z_2)$ to approximate $\varphi_x(z_1, z_2)$ over a given rectangle CR_i, and then using a composite function pl to approximate $\varphi_x(z_1, z_2)$ over a larger rectangular region R. To show the validity of these pl_i's and pl, we need to establish two lemmas. In this appendix we give the proofs for these two lemmas.

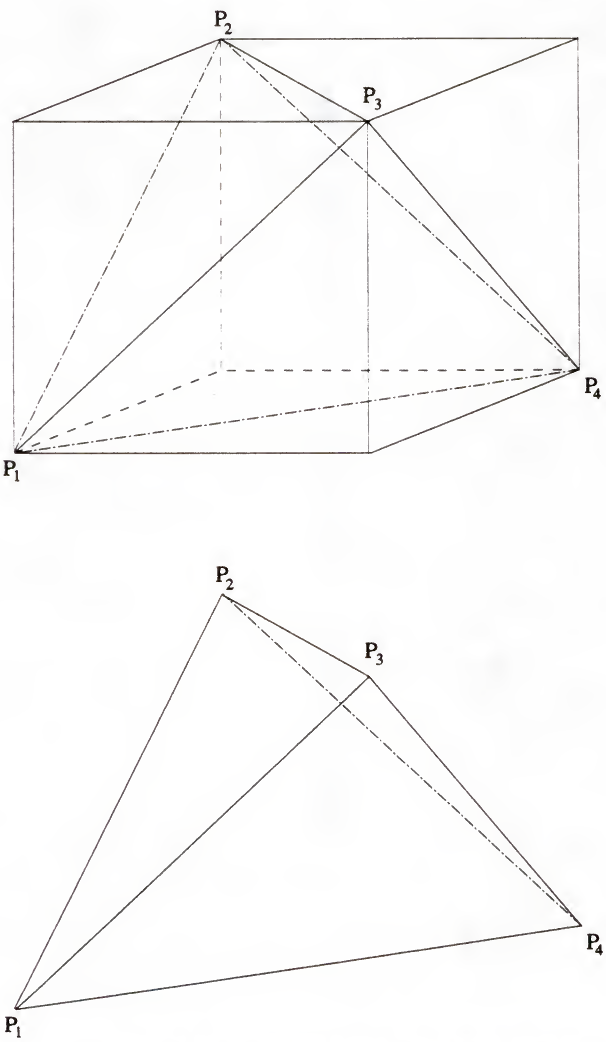


Figure A4.1 The Polytope inside a Cuboid

First, we emphasize that rectangle CR_i is a subregion of the square region SR_i where over the latter the grid network distance φ_x is also a 2-piecewise linear concave function. Over SR_i, function $\varphi_x(z_1, z_2)$ together with function $|z_1 - z_2|$ forms a polytope of four faces and four extreme points. It can be visualized as a geometric object carved out of a cube, as shown in Figure A4.1. Specifically, function φ_x over SR_i consists of the two faces $P_1P_2P_3$ and $P_2P_3P_4$, and function $|z_1 - z_2|$ consists of the two faces $P_1P_2P_4$ and $P_1P_3P_4$.

Lemma 4.3. $pl_i(z_1, z_2) = l_{i1}(z_1, z_2), \forall(z_1, z_2) \in C_{i1}$ and $pl_i(z_1, z_2) = l_{i2}(z_1, z_2), \forall(z_1, z_2) \in C_{i2}$.

Proof. From Figure 4.12 and Figure A4.2, we see that l_{i1} over C_{i1} is a two-dimensional right triangle in E^3 with extreme points $A = (z^{SW}, \varphi_x(z^{SW}))$, $B = (z^{NW}, \varphi_x(z^{NW}))$, and $C = (z^{NE}, \varphi_x(z^{NE}))$. Similarly, l_{i2} over C_{i2} is a two-dimensional right triangle in E^3 with extreme points A, C, and $D = (z^{SE}, \varphi_x(z^{SE}))$. Let Δ_1 and Δ_2 denote these two triangles respectively and let s be the piecewise linear surface that consists of these two triangles with their hypotenuses joined. That is, $s(z_1, z_2) = l_{i1}(z_1, z_2)$ if $(z_1, z_2) \in C_{i1}$ and $s(z_1, z_2) = l_{i2}(z_1, z_2)$ if $(z_1, z_2) \in C_{i2}$. If s is convex, then, since l_{i1} and l_{i2} are the linear supporting planes of s, we will have $l_{i2}(z_1, z_2) \le s(z_1, z_2) = l_{i1}(z_1, z_2)$ for any $(z_1, z_2) \in C_{i1}$ and $l_{i1}(z_1, z_2) \le s(z_1, z_2) = l_{i2}(z_1, z_2)$ for any $(z_1, z_2) \in C_{i2}$. Thus, it is sufficient to show that s is convex.

We prove this by considering all the possible positions of CR_i inside SR_i. Let $A_1 = \{(z_1, z_2) \in SR_i \mid z_1 + z_2 \le vl_i + vl_{i-1}\}$ and $A_2 = \{(z_1, z_2) \in SR_i \mid z_1 + z_2 > vl_i + vl_{i-1}\}$ (They are, respectively, the lower left and upper right triangles of SR_i). Rectangle CR_i can have the following positions:

i. $CR_i \subset A_1$;

ii. $CR_i \subset A_2$;

iii. $z^{NE} \in A_2$ and the other three corner points are in A_1;

iv. $z^{SW} \in A_1$ and the other three corner points are in A_2;

v. $z^{NE}, z^{SE} \in A_2$ and $z^{NW}, z^{SW} \in A_1$;

vi. $z^{NE}, z^{NW} \in A_2$ and $z^{SE}, z^{SW} \in A_1$;

Figures A4.2, A4.4, A4.5, and A4.7 give the positions of CR_i for cases iii, ..., vi, respectively.

Since φ_x is symmetric in A_1 and A_2, we only need to consider cases i, iii, v, and vi. Before discussing each individual case, we observe that, since $l_{i1}(z_1, z_2)$ $(l_{i2}(z_1, z_2)) \le \varphi_x(z_1, z_2)$ for any $(z_1, z_2) \in C_{i1}$ (for any $(z_1, z_2) \in C_{i2}$), we have $s(z_1, z_2) \le \varphi_x(z_1, z_2)$ for any $(z_1, z_2) \in CR_i$.

Case i. Since Δ_1 and Δ_2 are the same plane, s is linear.

Case iii. Consider Figure A4.2. Let $l(.\,,.)$ be the two dimensional plane in E^3 defined by points A, B, and D. Plane $l(.\,,.)$ coincides with φ_x over A_1, so that $l(z_1, z_2) \geq \varphi_x(z_1, z_2) \geq s(z_1, z_2)$ for any $(z_1, z_2) \in CR_i$. We can see from Figure A4.3 that, over domain CR_i, $s(z_1, z_2)$ consists of faces ABC and ADC, and $l(.\,,.)$ is the linear surface ABDE, where $E = (z^{NE}, l(z^{NE}))$. Surface s has four extreme points A, B, C, and D, so that s shares three common extreme points A, B, and D with linear surface ABDE. This situation can occur only when $s(z_1, z_2)$ is convex (If s were concave, then linear surface ABDE and surface s could not have shared extreme points B and D at the same time, given that they share extreme point A.)

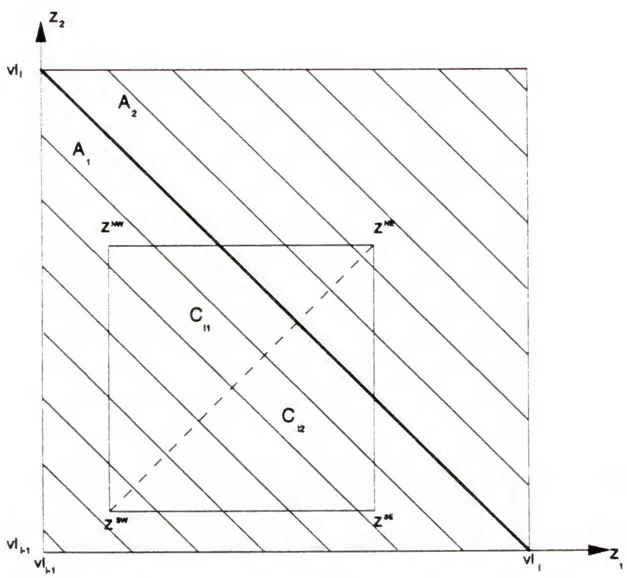


Figure A4.2 The Position of CR_i: Case iii

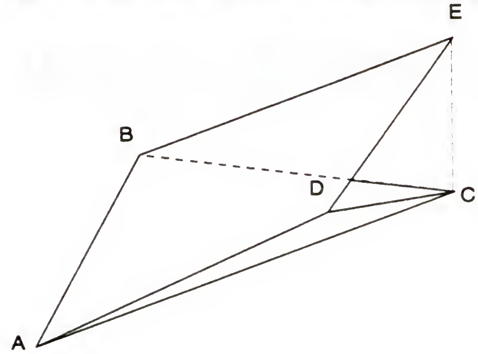


Figure A4.3 The Piecewise Linear Surfaces: Case 1

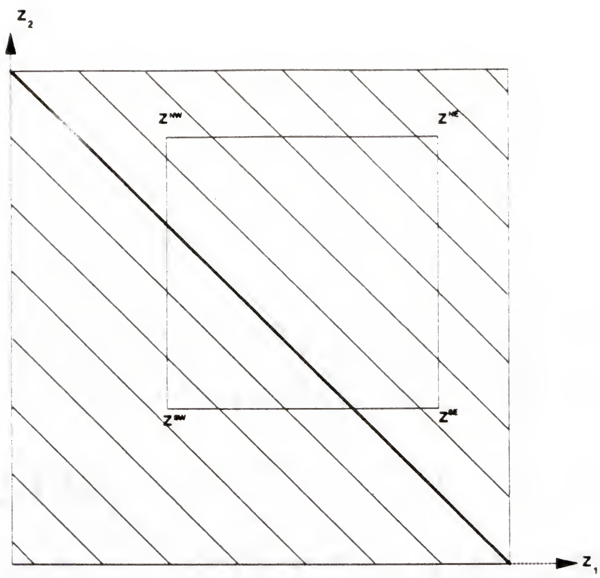


Figure A4.4 The Position of CR_i: Case iv

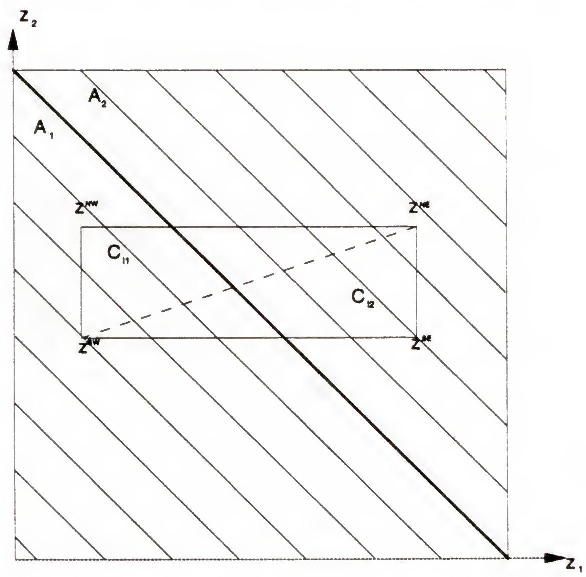


Figure A4.5 The Position of CR_i: Case v

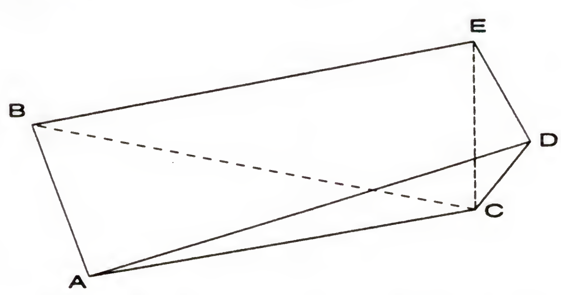


Figure A4.6 The Piecewise Linear Surfaces: Case 2

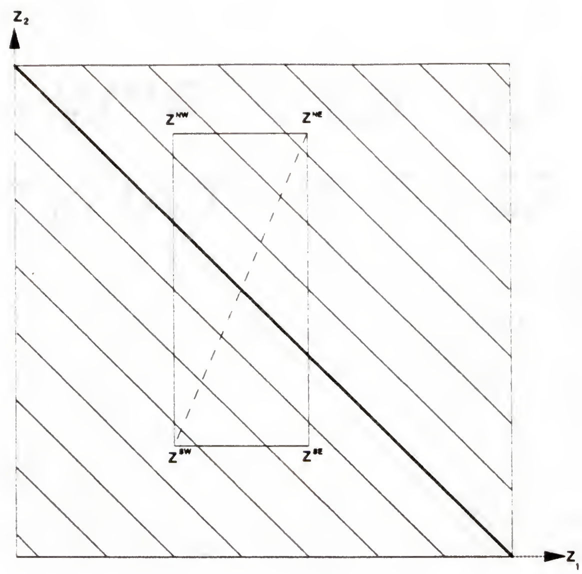


Figure A4.7 The Position of CR_i: Case 4

Case v. Consider Figure A4.5. Again, let $l(.\,,.)$ be the linear surface, in E^3, defined by points A, B, and D. Since $\varphi_x(z^{NW}) > \varphi_x(z^{SW})$ and $\varphi_x(z^{SE}) > \varphi_x(z^{NE})$, we have $l(z^{NE}) > \varphi_x(z^{NE})$. The three-dimensional images of l and s over CR_i are given in Figure A4.6, where l is the linear surface ABDE and s consists of faces ABC and ACD. Since over CR_i the 2-piecewise linear surface s is below the linear surface l and they share three of the four extreme points, s must be convex.

Case vi. This case is similar to Case v. Consider Figure A4.7. Here, we use that fact that $\varphi_x(z^{NW}) > \varphi_x(z^{NE})$ and $\varphi_x(z^{SE}) > \varphi_x(z^{SW})$. ■

Lemma 4.4. $pl_i(z_1, z_2) \leq |z_1 - z_2|$, for any $(z_1, z_2) \in R - CR_i$.

Proof. We prove this lemma by considering all possible positions of CR_i in R.

Case 1. $CR_i = R$

In this case, $R - CR_i = \varnothing$.

Case 2. $CR_i = SR_i$

From Theorem 4.1a, $pl_i(z_1, z_2) = |z_1 - z_2|$ for any (z_1, z_2). Thus, the inequality is true.

Case 3. CR_i is the "southwest" corner of R.

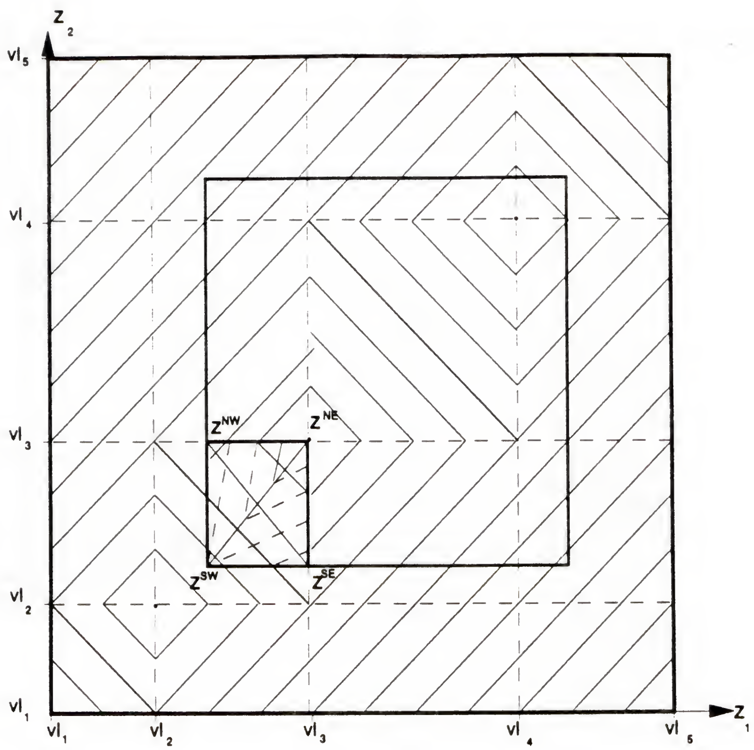


Figure A4.8 The Position of CR_i inside R: For Case 3 of Lemma 4.4

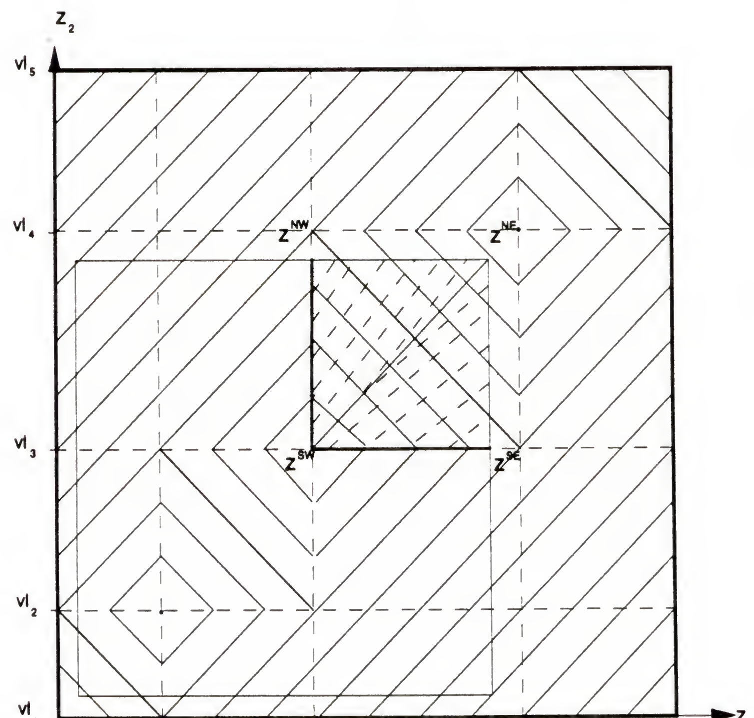


Figure A4.9 The Position of CR_i inside R: For Case 4 of Lemma 4.4

Figure A4.8 shows the position of CR_i in R. For this case, the intersection of pl_i and $|z_1 - z_2|$ is a right angled curve in E^3 passing through points $(z^{NW}, \varphi_x(z^{NW}))$, $(z^{NE}, \varphi_x(z^{NE}))$, and $(z^{SE}, \varphi_x(z^{SE}))$. The projection of this curve on the (z_1, z_2) plane separates CR_i from $R - CR_i$. Since, from Theorem 4.1a, we know that $pl_i(z_1, z_2)$ is above $|z_1 - z_2|$ over region CR_i, it must be below $|z_1 - z_2|$ over region $R - CR_i$.

Case 4. CR_i is the "northeast" corner of R

Figure A4.9 shows the position of CR_i in R. In this case, the intersection of pl_i and $|z_1 - z_2|$ has its projection on (z_1, z_2) plane as a right angled curve passing points z^{NW}, z^{SW}, and z^{SE}. The rest of the proof is similar to that of Case 3. ■

Appendix B.5 A Notation Glossary

G: A network (it usually refers to a cyclic network)

N_g: A grid network

m: number of vertices in G (N_g)

n: number of new facilities

G^n: The n-fold Cartesian product of G

$N_g{}^n$: The n-fold Cartesian product of N_g

PLC: Piecewise Linear and Convex

Notation for Multifacility Problems on a Cyclic Network G

X: Location variable vector $(x_1, \ldots, x_n) \in G^n$

P: Minimize $\{f(X) = c(D(X)) \mid X \in G^n\}$

P': A subproblem of P such that $X \in S$ with

S: A subset, call L-set, of G^n (It is a set of solutions defined by a series of lower and upper bounds for location variables and some inequalities each involving two location variables.)

Ω: a partition of G^n such that on each of its elements D(X) is linear

$CL_{[j]}$: A segment of some edge $e_{[j]}$ with index [j] related to x_j

$L_{[j]}$: A segment in some edge $e_{[j]}$ (it refers to a sub-segment in $CL_{[j]}$)

v_i^p: An antipodal point of v_i on edge e_p (a local maximum of distance $d(v_i, x)$ over e_p)

$e_{[j]}, e_{[k]}$: Edges of G, to which x_j and x_k are restricted

$u_{[p]}, w_{[p]}$: End points of e_p

$u_{[q]}, w_{[q]}$: End points of e_q

$u_{[q]}^p$: the antipodal point of $u_{[q]}$ on edge e_p

$w_{[q]}^p$: the antipodal point of $w_{[q]}$ on edge e_p

$u_{[p]}^q$: the antipodal point of $u_{[j]}$ on edge e_q

$w_{[p]}^q$: the antipodal point of $w_{[j]}$ on edge e_q

L_H: For the case $e_p \neq e_q$, L_H is line segment in $e_p \times e_q$ where the type-II distance d(. , .) reaches its maximum. The geometry of L_H is a line segment running 135 degrees inside rectangle $e_p \times e_q$

L_L: For the case $e_p = e_q$, L_L is line segment in $e_p \times e_q$ where the type-II distance d(., .) reaches its minimum. The geometry of L_L is a line segment running 45 degrees inside rectangle $e_p \times e_q$

H_{pq}: The hyperplane in $e_p \times e_q$ which coincides with L_H if $p \neq q$, and coincides with L_L if $p=q$.

$\{H_{pq}^-, H_{pq}^+\}$: H_{pq}^- and H_{pq}^+ are the mutual-complement half-planes such that $H_{pq}^- \cup H_{pq}^+ = e_p \times e_q$, and $H_{pq}^- \cap H_{pq}^+ = H_{pq}$

H: The collection of H_{pq} for every edge cross product

H^-: The collection of H_{pq}^-

H^+: The collection of H_{pq}^+

LR_{pq}: A linear region in $e_p \times e_q$. It is a subset of either H_{pq}^- or H_{pq}^+, obtained by adding lower and/or upper bounds for x_j and/or x_k

R_{jk}: For a given L-set S, $R_{jk} = \{(x_j, x_k) \mid X \in S\}$ (R_{jk} is a simple polytope of (x_j, x_k)

R_{jk}': A quadrilateral in E^2, which contains R_{jk} ($R_{jk}' = R_{jk}$ if R_{jk} itself is a quadrilateral, and $R_{jk} \subset R_{jk}'$ if R_{jk} is a pentagon)

Notation for Multifacility Problems on a Grid Network N_g

U, $u_j = (u_{xj}, u_{yj})$, $u_k = (u_{xk}, u_{xk})$: the location variables on N_g and their coordinates in E^2

U_x, U_y: $U = U_x \times U_y = (u_{x1}, \ldots, u_{xn}) \times (u_{y1}, \ldots, u_{yn})$

P: Minimize $\{f(X) = c(D(U)) \mid U \in N_g{}^n\}$

P': A subproblem of P such that $U \in S \cap N_g{}^n$ with

$S = S_x \times S_y$, where $S_x = \{U_x \mid lb_j \le u_{xj} \le rb_j, (u_{xj}, u_{xk}) \in X_{jk}\}$

$S_y = \{U_y \mid bb_j \le u_{yj} \le tb_j, (u_{yj}, u_{yk}) \in Y_{jk}\}$

X_{jk}: A simple polytope of (u_{xj}, u_{xk})

Y_{jk}: A simple polytope of (u_{yj}, u_{yk})

v-int vertex: a non-intersection vertex on a vertical grid line (in the interior of a grid row)

h-int vertex: a non-intersection vertex on a horizontal grid line (in the interior of a grid col)

$vl_1, \ldots, vl_p$: x-coordinates of vertical grid lines

$hl_1, \ldots, hl_q$: y-coordinates of horizontal grid lines

$vl_{[i]}, vl_{[i]}'$: x-coordinates of the vertical grid lines adjacent to vertex v_i

$hl_{[i]}, hl_{[i]}'$: y-coordinates of the horizontal grid lines adjacent to vertex v_i

$\Delta(a, b, c)$: A convex hull (a hyper-triangle) in E^p spanned by three linearly independent points a, b, and c

ϕ: A function on E^1 which is used to describe part of a shortest distance function $d(v_i, u_j)$

$$\phi(z|a_1, a_2, a_3) = \max\{|z - a_1|, \pi(z|a_1, a_2, a_3)\}, \text{ where}$$

$$\pi(z|a_1, a_2, a_3) = \min\{a_1 + z - 2a_2, 2a_3 - a_1 - z\}$$

δ_x, δ_y: functions on E^2 such that $d(v_i, u_j) = \delta_x(v_{xi}, u_{xj}) + \delta_y(v_{yi}, u_{yj})$

$$\delta_x(v_{xi}, u_{xj}) = \begin{cases} |v_{xi} - u_{xj}| & \text{if } v_{yj} = u_y = hl_k \text{ for some } k \\ \phi(u_{xj} \mid v_{xi}, vl_{[i]}, vl_{[i]}') & \text{o/w} \end{cases}$$

$$\delta_y(v_{yi}, u_{yj}) = \begin{cases} |v_{yi} - u_{yj}| & \text{if } v_{xj} = u_x = vl_k \text{ for some } k \\ \phi(u_{yj} \mid v_{yi}, hl_{[i]}, hl_{[i]}') & \text{o/w} \end{cases}$$

φ_x, φ_y: Functions on E^2 which are used to describe $\rho_x(u_{xj}, u_{xk})$ and $\rho_y(u_{yj}, u_{yk})$ respectively

$$\varphi_x(z_1, z_2) = \max\{|z_1 - z_2|, \tau_{x1}(z_1, z_2), \ldots, \tau_{x,p\text{-}1}(z_1, z_2)\}$$

$$\varphi_y(z_1, z_2) = \max\{|z_1 - z_2|, \tau_{y1}(z_1, z_2), \ldots, \tau_{v,q\text{-}1}(z_1, z_2)\},$$ where

$$\tau_{xi}(z_1, z_2) = \min\{z_1 + z_2 - vl_i, 2vl_{i+1} - z_1 - z_2\},\ i = 1, \ldots, p-1$$

$$\tau_{yi}(z_1, z_2) = \min\{z_1 + z_2 - hl_i, 2hl_{i+1} - z_1 - z_2\},\ i = 1, \ldots, q-1$$

ρ_x, ρ_y: functions on $E^2 \times E^2$ such that $d(u_j, u_k) = \rho_x(u_{xj}, u_{xk}) + \rho_y(u_{yj}, u_{yk})$

$$\rho_x(u_{xj}, u_{xk}) = \begin{cases} |u_{xj} - u_{xk}| & \text{if } u_{yj} = u_{yk} = hl_i \text{ for some i} \\ \varphi_x(u_{xj}, u_{xk}) & \text{o/w} \end{cases}$$

$$\rho_y(u_{yj}, u_{yk}) = \begin{cases} |u_{yj} - u_{yk}| & \text{if } u_{xj} = u_{xk} = vl_i \text{ for some i} \\ \varphi_y(u_{yj}, u_{yk}) & \text{o/w} \end{cases}$$

REFERENCES

Aho, A. V., J. E. Hopcroft, and J. D. Ullman. 1976. *The Design and Analysis of Computer Algorithms*, Addison-Wesley Publishing Co., Reading, MA, 1976.

Aikens, C. H. 1985. "Facility Location Models for Distribution Planning," Eur. J. Opnl. Res., 22, 263-279.

Bandelt, H-J. 1989. "Single Facility Location on Median Networks," Working Paper, Fachbereich Mathematik, Universitat Oldenburg, Oldenburg, F.R.G.

Batta, R., and S. S. Chiu. 1989. "Formalizing the Connection Between Network and Planar Manhattan Metric Single Facility Location Problems," Working Paper, Department of Industrial Engineering, State University of New York at Buffalo, Buffalo, NY. 14260.

Batta, R., A. Ghose, and U. S. Palekar, 1989. "Locating Facilities on the Manhattan Metric with Arbitrarily Shaped Barriers and Convex Forbidden Regions," Trans. Sci., 23(1), 26-36.

Brandeau, M. L., and S. S. Chiu. 1989. "An Overview of Representative Problems in Location Research," Mgmt. Sci., 35(6), 645-674.

Cabot, A. V., R. L. Francis, and M. A. Stary. 1970. "A Network Flow Solution to a Rectilinear Distance Facility Location Problem," AIIE Trans., 2, 132-141.

Chhajed, Dilip, and T. J. Lowe. 1992. "m-Median and m-Center Problems With Mutual Communication: Solvable Special Cases," Opns. Res., 40 (Suppl 1), S56-S66.

Chang, G. J., and G. L. Nemhauser. 1982. "R-Domination on Block Graphs," Opns. Res. Lett., 1(6), 214-218.

Chen, M. L., R. L. Francis, J. F. Lawrence, T. J. Lowe, and S. Tufekci. 1985. "Block-Vertex Duality and the One-Median Problem," Networks, 15, 375-412.

Chen, M. L., R. L. Francis, and T. J. Lowe. 1988. "The 1-Center Problem: Exploiting Block Structure," Trans. Sci., 22(4), 259-269.

Dearing, P. M., and R. L. Francis. 1974. "A Network Flow Solution to a Multifacility Minimax Location Problem involving Rectilinear Distances," Trans. Sci., 8(3). 333-343

Dearing, P. M., R. L. Francis, and T. J. Lowe. 1976. "Convex Location Problems on Tree Network," Opns. Res., 24, 628-642.

Egbelu, P. J. 1982. "A Design Methodology for Operational Control Elements for Automatic Guided Vehicle Based Material Handling System," Ph.D Dissertation, Dept. of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg.

Erkut, E., R. L. Francis, and T. J. Lowe. 1988. "A Multimedian Problem with Interdistance Constraints," Environment and Planning B, 15, 181-190.

Erkut, E., R. L. Francis, T. J. Lowe, and A. Tamir. 1989. "Equivalent Mathematical Programming Formulations of Monotonic Tree Network Location Problems," Opns. Res., 37(3), 447-461.

Erkut, E., R. L. Francis, A. Tamir. 1992. "Distance-Constrained Multifacility Minimax Location Problems on Tree Networks," Network, 22, 37-54.

Francis, R. L., T. J. Lowe, and H. D. Ratliff. 1978. "Distance Constraints for Tree Network Multifacility Location Problems," Opns. Res., 26(4), 570-596.

Francis, R. L., L. F. McGinnis and J. A. White. 1983. "Locational Analysis," Eur. J. of Opns. Res., 12, 1983, 220-252.

Francis, R. L., L. F. McGinnis, and J. A. White. 1992. *Facility Layout and Location*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Ghosh, Avijit, and G. Rushton (Ed). 1987., *Spatial Analysis and Location-Allocation Models*, Van Nostrand Reinhold Company, New York.

Goetz, W. G. Jr., and P. J. Egbelu. 1989. "Guide Path Design and Location of Load Pick-up/Drop-off Points for an Automated Guided Vehicle System," Intl. J. of Prod. Res., 12, 928-941.

Goldman, A. J. 1969. "Optimum Locations for Centers in Network," Tran. Sci., 3, 352-360.

Goldman, A. J. 1971. "Optimal Center Location in Simple Networks," Trans. Sci., 5, 212-221.

Goldman, A. J. 1972. "Minimax Location of a Facility in a Network," Trans. Sci., 6, 352-360.

Goldman, A. J., and C. J. Witzgall. 1970. "A Localization Theorem for Optimal Facility Placement," Trans. Sci., 4, 406-409.

Gurevich, Y.,L. Stockmeyer and U. Vishkin. 1984. "Solving NP-Hard Problems on Graphs That are Almost Trees and An Application to Facility Location Problems," J. ACM., 31, 459-473.

Hakimi, S. L. 1964. "Optimal Locations of Switching Centers and the Absolute Centers and Medians of a Graph," Opns. Res., 12, 450-459.

Hakimi, S. L.. 1965. "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems," Opns. Res., 13, 462-475.

Hakimi, S. L., and S. N. Maheshwari. 1972. "Optimal Location for Centers in Networks," Opns. Res., 20, 967-973.

Handler, G. Y., and P. B. Mirchandani. 1979. *Location on Networks Theory and Algorithms*, The MIT Press, Cambridge, MA.

Hansen, P., and J. F. Thisse. 1983. "Recent Advances in Continuous Location Theory," Working Paper, Estratto da SISTEMI URBANI," Rivista Quadrimestrate di scienze della citta edel territorio, Anno V. -N.1.

Harary, F. 1969. ***Graph Theory***, Addison Wesley, Reading, MA.

Hedetniemi, S. M., S. T. Hedetniemi and P. J. Slater. 1982. "Centers and Medians of $C_{(N)}$-Trees," Utilitas Mathematica, 21c, 225-234.

Hedetniemi, S. T., R. Laskar and J. Pfaff. 1986. "A Linear Algorithm for Finding a Minimum Dominating Set in a Cactus," Discre. Appl. Math., 13(3), 287-292.

Hooker, J. N. 1986. "Solving Nonlinear Single-Facility Network Location Problems," Opns Res., 34(5). 732-743

Hooker, J. N. 1989. "Solving Nonlinear Multiple-Facility Network Location Problems," Networks, 19(1). 345-355

Hooker, J. N., R. S. Garfinkel, and C. K. Chen. 1991. "Finite Dominating Sets for Network Location Problems," Opns Res., 39(1), 100-118.

Horst, R. 1990. "Deterministic Methods in Constrained Global Optimization: Some Recent Advances and New Fields of Application," Naval Res. Logistic., 37, 433-471.

Hua, Lo-Keng, and others. 1962. "Applications of Mathematical Methods to Wheat Harvesting," Chinese Math., 2, 77-91.

Idrissi, H., O. Lefebvre, and C. Michelot. 1987 "Duality for Constrained Multifacility Location Problems with Fixed Norms and Applications," Working Paper, Campus Universitaire Mentmuzard, Faculli de Sciences Mirande, BP 138, 21004 Dijon Cedex.

Juel, H., and R. F. Love. 1976. "An Efficient Computational Procedure for Solving the Multifacility Rectilinear Facilities Location Problem," Opns. Res. Q., 27, 697-703.

Kariv, O., and S. L. Hakimi. 1979a. "An Algorithmic Approach to Network Location Problems. Part I. The Centers," SIAM J. Appl. Math., 37, 513-538.

Kariv, O., and S. L. Hakimi. 1979b. "An algorithmic Approach to Network Location Problems, II: The p-Medians," SIAM J. Appl. Math., 37(3), 539-560.

Karzanov, A. V. 1974. "Determining the maximal flow in a Network by the Method of Preflows, "Soviet Math. Dokl 15, 434-437.

Kim, T. U., T. J. Lowe, J. E. Ward, and R. L. Francis. 1989. "A Minimum Length Covering Subgraph of a Network," Ann. of Opns. Res., 18, 245-260.

Kincaid, R. K., and T. J. Lowe. 1990. "Locating an Absolute Center on Graphs That Are Almost Trees," Eur. J. of Opnl. Res., 44, 357-372.

Kiran, A. A. and B. C. Tansel. 1989. "Optimal Pickup Point Location on Material Handling Networks," Intl. J. Prod. Res., 27(9). 1475-1486.

Kolen, A. J. W. 1981. "Equivalence Between the Direct Search Approach and the Cut Approach to the Rectilinear Distance Location Problem," Opns. Res. 29, 616-620.

Kolen, A. J. W. 1982. "Location Problems on Trees and In the Rectilinear Plane," Stitchting Mathematisch Centrum, Amsterdam.

Kolen, A. J. W. 1986. Tree Network and Planar Rectilinear Location Theory, Centrum voor Wiskunde en Informatica, Amsterdam.

Krarup, J., and P. Pruzan. 1979. "Selected Families of Location Problems," Ann. Disc. Math., 5. 167-178.

Krarup, J., and P. Pruzan. 1983. "The Simple Plant Location Problem: Survey and Synthesis," Eur. J. of Opns. Res., 12, 36-81.

Larson, R. C. and G. Sadiq. 1983. "Facility Locations with the Manhattan Metric in the Presence of Barriers to Travel," Opns. Res. 31(4), 652-669.

Lefebvre, O., C. Michelot, and F. Plastria. 1991. "Sufficient Conditions for Coincidence in Minisum Multifacility Location Problems with a General Metric," Opns Res., 39(3), 437-442.

Love, R. F., J. G. Morris, and G. O. Wesolowsky. 1988. *Facilities Location, Models & Methods.* North-Holland, Amsterdam.

Mirchandani, P. B., and R. L. Francis. 1990, *Discrete Location Theory,* J. Wiley & Sons, Inc. Somerset, New Jersey.

Nieminen, J. 1988. "The Center and The Distance Center of a Ptolemaic Graph," Opns. Res. Lett., 7(2), 540-550.

Picard, J. C., and H. D. Ratliff. 1978. "A Cut Approach to the Rectilinear Distance Location Problems," Opns. Res., 26, 422-434.

Pierce, J. F., and W. B. Crowston. 1971. "Tree Search Algorithms for Quadratic Assignment Problems," Naval Res. Logistic. Quart., 18, 1-36.

Pritsker, A. A. B., and P. M. Chare. 1970. "Locating New Facilities with Respect to Existing Facilities," AIIE Trans., 2, 290-297.

Proskurowski, A. 1980a. "Centers of Maximal Outerplanar Graphs," J. Graph Theory, 4, 75-79.

Proskurowski, A. 1980b. "Centers of Two-Trees," Annals of Discrete Math., 9, 1-5.

Rao, M. R. 1973. "On the Direct Search Approach to the Rectilinear Facilities Location Problem," AIIE Trans. 5, 256-264.

Rado, Francisc. 1988. "The Euclidean Multifacility Location Problem" Opns Res., 36(3), 485-492.

Rosentiel, P., J. R. Fiksel, and A. Holliger. 1972. "Intelligent Graphs: Network of finite Automata Capable of Solving Graph Problems," *Graph Theory and Computing*, R.C. Read (Ed.), Academic Press, New York.

Sherali, H. D., and C. M. Shetty. 1978. "A Primal Simplex Based Solution Procedure for the Rectilinear Facilities Location Problem," J. Opns. Res. S., 29, 373-381.

Shetty, C. M., and R. W. Taylor. 1987. "Solving Large-Scale Linear Programs by Aggregation," Computers and Opns. Res., 14(5), 385-393.

Tamir, Arie. 1993. "Complexity Results for the p-Median Problem with Mutual Communication," Working Paper, Department of Statistics, Raymond and Beverly Sackler, Faculty of Exact Sciences, Tel Aviv University, Ramat-Aviv, Tel Aviv 69978, Israel

Tansel, B. C., R. L. Francis, and T. J. Lowe. 1980. "Binding Inequalities for Tree Network Location Problems with Distance Constraints," Tran. Sci., 14. 183-197

Tansel, B. C., R. L. Francis,, T. J. Lowe. 1983a. "Location on Networks: A Survey, Part I," Mgmt Sci., 29(4), 482-497.

Tansel, B. C., R. L. Francis,, T. J. Lowe. 1983b. "Location on Networks: A Survey, Part II," Mgmt Sci., 29(4), 498-511.

Tansel, B. C., and A. S. Kiran. 1988. "Optimum Storage Location in Flexible Manufacturing Cells," J. of Mgmt Sci., 7(2), 121-129.

Witzgall, C. 1964. "Optimal Location of a Central Facility: Mathematical Models and Concepts," National Bureau of Standards Report 8388.

BIOGRAPHICAL SKETCH

Yang Xu was born on June 28, 1960, in Beijing, China. In 1978, he attended the People's University of China, where he received the Bachelor of Science and Master of Engineering degrees in computer science and information systems, in August 1982 and July 1985. In 1985, he married Weili He. In January 1986, he enrolled in the Industrial and Systems Engineering Department of the University of Florida.

I certify that I have read this study and that in my opinion it conforms to acceptable standard of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Richard L. Francis

R. L. Francis, Chairman
Professor of Industrial and Systems Engineering Department

I certify that I have read this study and that in my opinion it conforms to acceptable standard of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Harold P. Benson

H. P. Benson
Professor of Decision and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standard of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D. Jack Elzinga

D. J. Elzinga
Professor of Industrial and Systems Engineering Department

I certify that I have read this study and that in my opinion it conforms to acceptable standard of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

D. W. Hearn
Professor of Industrial and Systems Engineering Department

I certify that I have read this study and that in my opinion it conforms to acceptable standard of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

C. Y. Lee
Associate Professor of Industrial and Systems Engineering Department

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partially fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1993

Winfred M. Phillips
Dean, College of Engineering

Dean, Graduate School